Review of social media traffic at the DNS resolvers and their security implementation.

ROBIN KREUGER, University of Twente, The Netherlands

Abstract: The Domain Name System (DNS) is meant to make our lives easier, by giving names to IP addresses similar to the old phone books. Even though security extensions like DNSSEC are widely implemented, not all systems seem secure. This research will use real traffic logs of links posted on social media to analyze how the traffic behaves of big cooperate social media providers in particular: WhatsApp, Telegram, Reddit, Slack, and LinkedIn. Since social media platforms are kind of a black box, it is hard for the end user to detect whether the links they have sent are indeed secure. Especially when the query does not originate from the end user's device, but rather from the servers of the social media providers. By analyzing the DNS(SEC) traffic, we concluded that certainly not all providers are secure in an attack scenario.

Additional Key Words and Phrases: Security, Traffic, Public Key Cryptography, Social Media

1 INTRODUCTION

Nowadays many protocols exist to make our online behaviour safer, by adding confidentiality, authenticity and integrity. However, their presence does not directly mean that they are also implemented correctly or at all [3, 21]. Many social media platforms and their apps appear to their users as if they were a closed system. In reality, however, many of the user's interactions still rely on core Internet protocols, like the DNS. Vulnerabilities in these protocols could unknowingly risk the security and privacy of the users if they are not implemented correctly by the platform. Whenever a link to a website like google.com gets sent through a messenger a preview card gets loaded in the foreground. But also a lot is happening in the background that the average end user has no idea of. The Domain Name Service translates google.com to an address computer can understand. The older DNS protocol is quite vulnerable to spoofing, tunnelling, amplification, and more. Luckily a protocol extension has been introduced in 1994[22] called DNNSEC. Even though this extension has been around for almost 30 years, only half of all queries make use of DNSSEC[2, 3, 17, 19].

Social media as a research field was chosen due to two reasons: Even though social media often is a closed system[1], DNS queries are made when the web crawler tries to load a preview card, potentially leaving a user vulnerable, especially with providers ever-changing structure [4], which could introduce new security risks. Secondly, social media is used a lot among the youth [5], which is a group who is often unknowingly abused through social media[13, 15].

The focus of this paper is to investigate the role of DNS in user interactions on social media. To answer this question, we formulated four sub-questions:

- **RQA** Which interaction initiates a DNS query?
- **RQB** From which client does the DNS query originate; the viewer, the poster, or the social media provider?
- **RQC** If the query originates directly from the social media provider: Is the query protected against manipulation or eavesdropping?
- **RQD** How do social media platforms behave if the DNS query gets resolved in an attack scenario?

To answer these questions a qualitative analysis will be performed, by setting up a dedicated dummy website as well as an own authoritative name server, giving full access to all traffic logs as well as being in control of all parameters, regarding certificates.

2 BACKGROUND

To allow the reader to fully understand the following results a bit of background knowledge about how DNS and DNSSEC work is needed. This section will summarize the most important aspects of DNS, DNSSEC, and the basic interactions of social media messengers.

2.1 DNS

Each website that is online visible is located on some sort of server. These servers are located at a certain IP Address, which is similar to the addresses of houses in the real world. IP Addresses can be of two types: IPv4 which looks like this: *130.89.163.14*, or the newer IPv6 (*2001:67c:2564:331:f138:127c:b417:c9fd*). As it is quite hard to remember all of these numerical combinations in 1985 the Domain Name System (short DNS) was introduced, to give IP Addresses corresponding names like *google.com*. As there are millions of different domain names the system needs a bit of a structure, but first let's have a look at a domain name from an internet point of view: Consider the following link: *https://blog.rkreuger.nl./home* . Going through it from left to right: https://: the protocol, blog.: the hostname label, rkreuger.: the domain name label, nl: the top-level domain (TLD) label, .null: the root label, /home: the path. Figure 1 depicts how a query coming in at the resolver deals with the different labels[2]:

(1) The end user looking up the URL?
(2) The query gets send to recursive resolver (rr), asking for the IP Address of rkreuger.nl?
(3) The rr asks the root server where .nl TLD server is located?
(4) The root server responds with a list of .nl TLD nameserver, including their respective IP Addresses!
(5) The rr uses that IP Address to ask the .nl TLD server where rkreuger.nl is located?

[1] e.g. the end user only interacts with the messenger app

[2] The colours correspond to the text

TScIT 38, Febuary 3, 2023, Enschede, The Netherlands

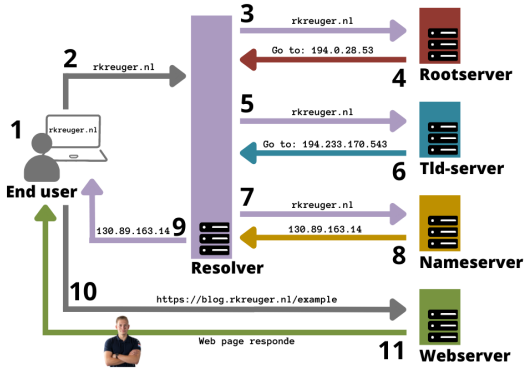

Fig. 1. Traffic flow at a recursive DNS resolver.

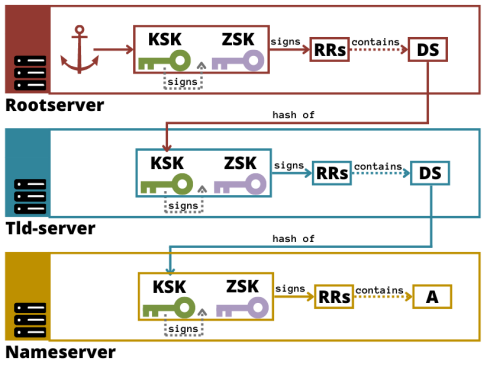

Fig. 2. Depiction of root zone's chain of trust [22].

(6) The .nl TLD server responds with a list of rkreuger.nl nameserver!
(7) The rr uses that IP Address to ask the domain name server of rkreuger.nl where it is located?
(8) The rkreuger.nl nameserver responds with the final IP Address!
(9) The rr responds to the end user with the IP Address of rkreuger.nl!
(10) The end user can now start fetching content from blog.rkreuger.nl/example?
(11) The webserver at blog.rkreuger.nl replies with the content of the /example page!

2.2 DNSSEC

As the standard DNS protocol does not allow for any validation of correct results (e.g. rkreuger.nl actually is at 130.89.163.14), security extensions like DNSSEC were added later. [22] Let's consider the following excerpt of the zone of rkreuger.nl:

(1) `rkreuger.nl A 130.89.163.14`
(2) `rkreuger.nl AAAA 2001:67c:2564:...:b417:c9fd`
(3) `rkreuger.nl NS ns1.rkreuger.nl`
(4) `rkreuger.nl NS ns2.rkreuger.nl`

This "plain" DNS zone contains 4 records. The first record `A`, defines where the IPv4 address of rkreuger.nl is located, and the second record `AAAA` where the IPv6 address is located. The last two are nameserver records, which tell the resolver which nameservers are responsible for handling the query. The first step in the DNSSEC validation, each record needs to be signed. For each domain there exists at least one pair of private and public keys, where the private key - as it indicates - is kept private and the public key becomes part of the records (`DNSKEY`). The private key signs records, where the public key is used to verify the signature. These signatures are records in themselves and called `RRSIG`. As this key pair signs the zones, it is called the Zone-Signing-Key (`ZSK`). With the combination of the `DNSKEY`, public `ZSK` and the `A or AAAA` records the resolver can verify the correctness of the `A or AAAA` record.
However, the problem arises of trust: How do we know that the

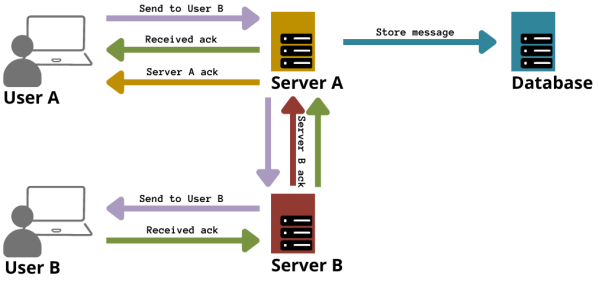

Fig. 3. Depiction of a general message flow.

`DNSKEY` is indeed correct? To validate it we can make use of a chain of trust (See Figure 2) of the root server[3]. The root server again has two pairs of keys, a ZSK pair and the second pair called the Key-Signing-Key (KSK) and together they are used to indirectly sign the KSK of the TLD server, by hashing the `DS` record. With this, the KSK and ZSK of the TLD server can be validated, and together they can validate the KSK and ZSK of the domain name server. As the signing can also be checked in reverse and the root server is a known trusted anchor we can assure that DNSKEY from above is indeed the right key for our RRSIG records and that our webpage is actually at the said location.[22]

2.3 Social Media

In addition to the DNS side of this research, it is also very useful to have a brought overview of how the traffic flow of a message posted in chat looks like. Based on Faang's article about messengers

[3]Colors correspond to Figure 1 and 2.

[6] Figure 3 shows the usual flow of messages. It starts from the message that is sent by *User A* towards *User B*. Once the message has been received by the server that manages the connection of user A an acknowledgement is returned, as well as the message is stored in the database. In addition, the message is also sent to User B handling server from where the message gets pulled by user B, which returns a received acknowledgement through server B, server A, and user A respectively, and finishes the message traffic until the next message is sent.

3 RELATED WORK

In general, this paper focuses on a very specific niche topic, hence not a lot of research is performed in this area. Nevertheless, social media in itself is widely researched and a couple of papers also look into the traffic of Facebook and WhatsApp. The hosting infrastructure between Facebook and WhatsApp differs a lot, even though both companies are part of Meta. Facebook makes use of a multitude of content distribution networks (CDN) spread all over geographical locations. Where WhatsApp is fully centralized in cloud hosting in the USA[1, 8]. Other work also showed inefficiencies in the transmitting of media content of social media networks[7], which shows that communication protocols are not at their optimum. Also recent discoveries of vulnerabilities at Threema[4] show[18], despite end-to-end encryptions, plenty of insecurities are still present in messenger applications.

4 METHODOLOGY

As there is a vast amount of different social media platforms, the decision of which to select was not easy. In the end, platforms were chosen with mainly a messenger function, therefore platforms like YouTube, and TikTok were out of the game, as they lack this feature, even though they have a market share of roughly 80%[5]. Also, not all platforms show a preview card when a link is posted, like Instagram. Facebook did meet all of the above criteria and is still widely used (69%[5] [5]), however, is strongly decreasing among the younger adults[5]. In addition, plenty of research has been done regarding Facebook's services, hence the focus on smaller providers is desired. By looking at each of the market shares of WhatsApp, Reddit, LinkedIn, Slack, and Telegram, with 23%, 18%, 28%, 3%, and <3% respectively [5, 9, 10], a selection was made of larger and smaller providers. Which combines end-to-end encryption methods, professional platforms as well as pleasure entertainment. In this paper, we will be analyzing how a user interacts, and initiates a DNS query, by analyzing the HTTP(s) requests that come in at the webserver. We will also be looking at the way social media platforms deal with domain name lookups, by not signing, signing, and incorrectly signing the queries. As the server handling differs quite a lot between the social media providers and is a bit like a black box, not giving a lot of insights into how the website links are being handled, the decision was made to set up an own authoritative nameserver to catch all of the DNS queries that are passing by as well as allowing us to purposely altering DNSSEC's zoning files, making them invalid and analyzing the provider's behaviour. Since TransIp[6] required two nameservers, each with a unique IP Address, bind was configured twice in a master slaves configuration and both servers were monitored during each case of the 3 cases: Plain DNS, correctly signed DNSSEC, and DNSSEC with broken signatures.

5 RESEARCH ANSWERS

In the following section, we focus on the quantitative analysis of answering the research questions. We will first be looking into the timing of DNS queries whether the sender, receiver, or both parties initiate a preview. After that, we will be looking further into where the queries originate from, from the users or the social media providers. Once we have an understanding of all the message flows we will be analyzing their vulnerabilities from a DNSSEC point of view and evaluating their security risks.

5.1 RQA - Which interaction initiates a DNS query?

In general, there are there possibilities where preview cards (e.g. Image, title, and description) can be loaded: Either at each user (e.g. *User A & User B*, see Figure 3) every time the message is being opened. Or once at *User A* when they type it in the chat and it then gets sent to *User B* containing the preview card, which would result in higher traffic throughput for the servers. Or the previews are being fetched at the servers themselves, where they are then sent to the users. The latter could result in privacy infringements, especially for WhatsApp as their main feature is end-to-end encryption, which means that their servers should not be able to read any messages including links. To assess where previews are being fetched the link to the dummy website *https://rkreuger.nl* was posted on the messenger/chat functions at each of the before mentioned social media platforms. Afterwards, the traffic at the website server was analyzed by using *"tshark"*[7]. By analyzing the incoming traffic, more specifically the traffic for the certain preview image, that is put on the dummy website, the following Table 1, was created based on the traffic logs.

Table 1. Overview of preview initiation location per social media.

When	Sender	Receiver
Whatsapp	Yes	No
Reddit	Yes	Yes
Slack	Yes	Yes
Telegram	Yes	No
LinkedIn	Yes	No

As can be seen in Table 1 there are two different scenarios of who loads the preview: In the case of WhatsApp, Telegram and LinkedIn the messages are being loaded by the sending user and sent with the message to the receiver, where Reddit and Slack request the preview at both the sending as well as the receiving user.

[4] A paid, open source, and secure communication platform.
[5] Market share across all adults.
[6] A domain name registration service. Necessary for the glue records.
[7] https://www.wireshark.org/docs/man-pages/tshark.html

5.2 RQB - From which client does the DNS query originate?

Based on the findings at RQA, further analyses were made, about the origin of the queries. Log files[8] were captured for each of the social media providers and the corresponding IPv4 addresses can be seen in Table 2. For each of the addresses, also the *WHOIS* (using whatismyip.com) data was queried in a second step to get more insights into the location and owner of the corresponding servers.

Table 2. Overview of preview retrieval location per social media.

Where	IPv4 Location	Country	Server
Whatsapp	end-users IP	end-users Country	none
Reddit	54.221.198.105	USA, VA, Ashburn	AWS
Slack	3.236.11.34	USA, VA, Ashburn	AWS
Telegram	149.154.161.245	GB, Warrington	Telegram
LinkedIn	52.143.249.184	USA, IA, Des Moines	Microsoft

From the above results three things are worth mentioning: WhatsApp is the only one that queries the preview through the user's connection, meaning that the user makes the request rather than WhatsApp's servers, which is also a requirement for WhatsApp's end-to-end encryption. Since WhatsApp performs its queries in this way, the exact server location of the provider can also not be retrieved by the form of this research method. However, by analysing the end-users IP of the HTTP request through a service like MaxMind we could still get an indication of the rough city. Nevertheless, as WhatsApp is owned by Facebook/Meta its servers are also very likely located in the USA. The second noticeable result is that Telegram is the only provider out of four, which has its servers within Europe, meaning they have to stick to the GDPR, which is not the case for servers that are located in the USA. The general data protection regulations tend to ensure privacy better, as well as protect the users against abuse of their data. Especially DNS resolvers have a privileged role in our internet traffic, hence transparency and privacy are hugely important and strictly regulated under GDPR, but unfortunately still lacking[11]. The last eye-catching trend is visible in the form of cloud computing, 3 out of the 4 known servers make use of either AWS's or Microsoft's cloud computing platforms, meaning the data does not only exists on the servers of social media providers themselves but is also located at thirds. As none of the providers lets the receiving users make the query themselves, the chance of the user's location leaking gets reduced a lot, as their IP address does not appear, but rather one of the social media providers. However, in the case of Reddit and Slack, as both providers do not allow the sending user to see whether a message was read by the receiving user, a desperate sending user could know whether the receiving user has read the message, by logging the traffic and filtering on the known ranges of the respective providers.

[8]By using *tshark -t u | tee <logfilename>*

5.3 RQC - Is the query protected against manipulation or eavesdropping?

In the previous sections, we have focused more on the traffic in general, to understand how different providers deal with links in previews. In this section, we want to dive more in-depth into the DNS queries themselves, especially how they make use of the security extensions of *DNSSEC*. Resolving bogus signed records, in the case of spoofing and cache poising, gives the attacker full control of what a user is seeing, including but not limited to capturing credentials and other important information.
In order to analyse the traffic, 2 authoritative nameservers were setup at *ns1.rkreuger.nl* and *ns2.rkreuger.nl* (see Appendix A). To see any effects and allow us to evaluate the results correctly, 3 tests were performed: plain DNS, without the DNSSEC extension, enable; with DNSSEC configured correctly; and by purposely falsifying the zoning files of DNSSEC, simulating an attack. The results of this can be seen in Table 3 and 4 respectively below.

Table 3. Overview of providers' possibility to fetch previews.

Provider	Plain DNS	DNSSEC Signed
Whatsapp	Yes	Yes
Reddit	Yes	Yes
Slack	Yes	Yes
Telegram	Yes	Yes
LinkedIn	Yes	Yes

Table 4. Overview of providers' reaction to a possible attack, through spoofing and which payload could be sent.

Provider	Spoofing Attack	Payload
Whatsapp	Cache fallback	None
Reddit	2nd Image & Text	Title & Describtion
Slack	Yes	Full preview
Telegram	No	None
LinkedIn	Cache fallback	None

As of right now, only roughly 58% [3, 21] of all DNS queries make use of DNSSEC, hence nearly half of all queries still use plain DNS, which results in a security risk. Nevertheless, all of the social media providers still fetch the image, which is beneficial from a content-design point of view but leaves the content vulnerable to attacks like spoofing, where instead of the designated content a different payload is shown. One could argue that it does not matter at a social media preview, as no receiving users' content is being exchanged, however, it could be a first step into tricking a user into a larger attack.
For the column of DNSSEC, it is good to see, that when DNSSEC is implemented all social media are still able to fetch the previews. The results in the attack scenario divert a lot. Going through the results from "safest" results to most vulnerable: Telegram, is the safest, as it does not show any content with a bogus link, as it is supposed to. WhatsApp and LinkedIn are quite similar to Telegram, but rather than not showing anything at all they have decided to

opt more on the design choice priority since when there is a cached option available (e.g. a preview card for the same link was requested earlier) they show the older picture, which was loaded when the link was still safe. Since WhatsApp resolves the query through the end-users connection, it is dependent on the end-users resolver whether the domain name gets resolved. The problem that arises from this method, is that the end-user has nearly no chance of actually knowing that the preview is cached. In case of an attack scenario, a cached preview card could not alert the user in the best possible way. More about the tests regarding caching later. Reddit has a different very interesting approach to the problem of showing a preview of an attack scenario. On the dummy website (See Figure 4) in total 3 pictures are present: A primary portrait picture, a secondary random picture of the University of Twente, and a compressed version of the primary in the header meta tags[9]. The picture of the UT is intended to simply be random content, however, with the added benefit, that the image itself is not hosted on rkreuger.nl, but rather at utwente.nl[10], which still has an intact DNSSEC signature. The description of the preview card still gets loaded from the meta-tags at rkreuger.nl, making it very hard for the end user to detect a potential attack. And last but not least, Slack, which does nothing with invalid signatures, continues fetching the previews. Table 4 also shows which parts of the preview could be changed by a payload in a case of a spoofing attack. Unfortunately, only Telegram, WhatsApp and LinkedIn, did not show pieces of the payload in this experiment. Nevertheless, none of the providers showed a warning that the signature was bogus.

5.4 RQD - How do social media platforms behave, if the DNS query gets resolved in an attack scenario?

We saw in the previous section that nearly all providers still display a preview, even though the end user could not notice any effect when messaging a link. As shown in the previous section, some providers cache previously fetched results. To study the caching behaviour more thoroughly, we altered the preview card text as well as the image to a really obvious different content, for each of the 3 tests. To possibly answer the question of how long WhatsApp and LinkedIn cache the picture, the same test was performed again after a day as well as a week and the content was still cached. Also forcing the cache to update by sending the URL [11] directly in the chat did not have any effect. However, it showed a different interesting result (See Table 5). When posting a URL with a .png ending 4 out of the 5 providers still allowed for fetching of said image and did not fall back on a possible cache, even though the signature was bogus.
For WhatsApp it does not automatically mean that they would not show the image, since they are very strict with meta tags[12] for displaying preview cards, it would not directly mean that it blocks the URL itself, but rather refuses to load the preview, due to the missing meta tags in the headers (See Appendix A.2). Hence, it can be said that pictures are still vulnerable to attacks on most platforms.

[9]So not visible in the screenshot
[10]https://www.utwente.nl/.uc/i38b3dbca01023f5003003fbbd4032c2ef644a585b06e0701\c3dc05000080/utimg25200.jpg
[11]https://rkreuger.nl/static/images/ProfilePicturePreviewFacebookNew.png
[12]The image URL does not contain any meta tags

Table 5. Attack Image URL Fetching.

Provider	Preview of Image URL
Whatsapp	No
Reddit	Yes
Slack	Yes
Telegram	Yes
LinkedIn	Yes

6 DISCUSSION AND CONCLUSION

As already explained in Section 5.1, there are multiple possibilities of how a social media network can be designed. Each of the choices has its advantages and disadvantages. So it all comes down to the infrastructure, resources as well as requirements. Looking at whether it is better that both the sender and receiver fetch the preview or only the sender initiate the crawling, can be seen in two ways: when both users request content, then they should have more up-to-date content, which is especially useful for dynamic content and reduces the traffic which goes through the provider's server. In the second case, the sender could verify that the preview he wants to send looks what he wants it to look like and does not change with dynamic content changes when the receiver reads it. Personally, I would prefer a preview that copes with dynamic changes as it gives a better impression of what a webpage looks like, and the receiving user could be better alerted when the link has a bogus DNSSEC signature.
Further looking at where the query gets loaded: Surprisingly, WhatsApp was the only one, that loaded the preview through the client's connection rather than the provider's server. Although it was somewhat expected as it copes with their e2e requirement, otherwise it would infringe on the confidentiality of the messages. As the other four providers do not have this requirement, they have the freedom of also requesting the preview through their servers. I assumed that they also capture that data, for them to make a small profit by selling it to third parties, but was not able to prove it conclusively. Nevertheless, numerous articles describe how user data mining helps with personalising advertisement content[12, 20]. This seems logical from their perspective, however, is not beneficial for the user's privacy.
Another very difficult design choice arises when talking about DNSSEC in general: How should an end user[13] be made aware when zones are signed incorrectly? As DNS does not directly involve the actual content of a webpage, should the end user still be presented with the queried webpage and a warning or nothing at all? The current implementation at each platform makes it nearly impossible for the user to see that the signature is wrong. I would add an icon alerting the user about it, as it addresses both requirements of design[14] and security[15]
Among experts the discussion of whether providers should validate signatures at their resolvers is ambiguous[14, 16]: On the one hand, it could be beneficial if providers would redirect bogus signature records to an error page, really alerting the user that something

[13]Not only on social media but also browsers.
[14]Showing the content anyhow.
[15]Making the user alert that the link could be tempered with.

could be wrong. However, on the other hand, it would also take away a lot of freedom of a person visiting the site, since not all webpage admins can maintain all pages constantly, possibly resulting in more frequent failing signatures and downtime for the website. Personally, I would suggest a middle way, similar to how browsers are handling bogus SSL certificates, where an option is presented to accept the risk and continue browsing. How this should be implemented in web crawlers is another design challenge in itself [16]. Lastly, I would like to address a possibly very concerning fact, namely, even though WhatsApp is end-to-end encrypted, meaning that WhatsApp should not be able to read or store anything[16], it seems that their previews still get cached and it is probably not happening on the local device, as sending the image from a multitude of different devices as well as accounts[17] still showed the older cached picture. Certainly, further analysis is needed to look into this claim, as a dedicated ivestigation was outside of the scope.

7 FUTURE WORK

Of the 5 providers, only WhatsApp has end-to-end encryption enabled by default, however, also Telegram offers end-to-end encryption. Possible future work could involve comparing Telegram to WhatsApp, and analyzing how their protocol/methods differ from each other. It was already addressed a little bit at RQD, but the further investigation could be done by comparing dynamic against static content, especially laying focus on how the providers cache the content. Also, the exceptional caching of WhatsApp as was discussed in the previous section could be addressed there as well. Lastly, the research could include more messaging providers, like Facebook, Twitter, Snapchat, and more.

REFERENCES

[1] *2015 IFIP Networking Conference (IFIP Networking)*. IEEE, 2015.
[2] APNIC Labs. Use of DNSSEC Validation for World, 10 2022.
[3] apnic.net. Use of DNSSEC Validation for World (XA).
[4] Andreas Berger and Eduard Natale. Assessing the Real-World Dynamics of DNS. Technical report.
[5] BY Brooke Auxier and Monica Anderson. Social Media Use in 2021 FOR MEDIA OR OTHER INQUIRIES. Technical report, 2021.
[6] Crack Faang. Designing Facebook Messenger, 6 2021.
[7] Qian Deng, Zhenyu Li, Qinghua Wu, Chuan Xu, and Gaogang Xie. An empirical study of the WeChat mobile instant messaging service. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 390–395. IEEE, 5 2017.
[8] Reza Farahbakhsh, Angel Cuevas, Antonio M. Ortiz, Xiao Han, and Noel Crespi. How far is Facebook from me? Facebook network infrastructure analysis. 5 2017.
[9] Finance online. 54 Essential Slack Statistics: 2023 Market Share Analysis & Data, 2022.
[10] FinanceOnline. 100 Telegram Statistics You Must Know: 2023 Users, Security & Revenue Data, 2022.
[11] Vijay K Gurbani, Cynthia Hood, Anita Nikolich, and Henning Schulzrinne. When DNS goes dark: Understanding privacy and shaping policy of an evolving protocol. Technical report.
[12] Kim Hammar, Shatha Jaradat, Nima Dokoohaki, and Mihhail Matskin. Deep Text Mining of Instagram Data without Strong Supervision. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 158–165. IEEE, 12 2018.
[13] Lisa M. Jones, Kimberly J. Mitchell, and David Finkelhor. Trends in youth internet victimization: Findings from three youth internet safety surveys 20002010. *Journal of Adolescent Health*, 50(2):179–186, 2 2012.
[14] Siva Kesava Reddy Kakarla, Ryan Beckett, Behnaz Arzani, Todd Millstein, and George Varghese. GRooT. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 310–328, New York, NY, USA, 7 2020. ACM.
[15] Lies Kimpe, Michel Walrave, Koen Ponnet, and Joris Ouytsel. Internet Safety. In *The International Encyclopedia of Media Literacy*, pages 1–11. Wiley, 5 2019.
[16] Warren Kumari, Evan Hunt, Roy Arends, Wes Hardaker, and David C Lawrence. RFC 8914: Extended DNS Errors. Technical report, 2020.
[17] Meng Luo, Yepeng Yao, Liling Xin, Zhengwei Jiang, Qiuyun Wang, and Wenchang Shi. Measurement for encrypted open resolvers: Applications and security. *Computer Networks*, 213, 8 2022.
[18] Kenneth G Paterson, Matteo Scarlata, and Kien Tuong Truong. Three Lessons From Threema: Analysis of a Secure Messenger. Technical report.
[19] Pablo Rodriguez. PERCEPTIONS OF IT DECISION-MAKERS ON THE USE OF DOMAIN NAME SYSTEM SECURITY EXTENSION (DNSSEC): QUALITATIVE EXPLORATORY CASE STUDY. Technical report, 2020.
[20] M Saffari, M M Poursaeed, and A Niknafs. Factors Influencing Personal Branding on Social Networks (Instagram) with Data Mining Approach. Technical Report 2, 2021.
[21] SIDN. Domain names with DNSSEC , 1 2023.
[22] Olivier Van Der Toorn, Moritz Müller, Sara Dickinson, Cristian Hesselman, Anna Sperotto, and Roland Van Rijswijk-Deij. Addressing the challenges of modern DNS a comprehensive tutorial, 8 2022.

A A - TEST ENVIRONMENT

A.1 Appendix A.1 - The nameservers

For setting up the test environment two[18] nameservers were needed. For setting them up, two tutorials were especially useful: https://www.digitalocean.com/community/tutorials/how-to-setup-dnssec-on-an-authoritative-bind-dns-server-2 and https://www.sidn.nl/en/modern-internet-standards/dnssec-signatures-in-bind-named. To verify all steps a couple of tools proved to be very useful: As can be seen in Figures 6 and 7 dnsviz.net, as well as zonemaster.net, was used to quickly test how the zone file would like at the resolver. For testing, whether the updates were also correct at the local machine the `dig @localhost rkreuger.nl +multiline` command was very helpful. To debug certain records/signatures `SOA`, `DNSSEC`, or `A` could be added to the previous dig commando. Also, `tcpdump` and `tshark` helped to analyze the traffic in the end. In the end, the test environment was comparable to Figure **??**.

The following subsections are meant to give a better overview of each machine and how it was configured:

A.1.1 Appendix A1.1 - ns1.rkreuger.nl. This server was the starting point for setting up bind, as I own this server myself. For a nameserver, it is way too powerful, but it was free to use for me. The specs are the following:

- HP Proliant DL380 on Campus at University of Twente
- Fedora 32 Server Edition (CentOS)
- IPv4: 130.89.163.14
- IPv6: 2001:67c:2564:331:f138:127c:b417:c9fd
- Bind: 9.11.28-RedHat-9.11.28-1.fc32

A.1.2 Appendix A1.2 - ns2.rkreuger.nl. Whilst trying to register the first nameserver, I noticed that the domain environment of TransIP as well as Mijndomein required two unique IP Addresses to function. Therefore the smallest Linode cloud node was bought, which is very affordable (0.00075€/hour) and bind was configured there as a master, as it was easier. And therefore my other server became the slave. The specs are the following.

- Linode cloud node in Germany

[16] Except on the local device, e.g. cookies.
[17] I have a German number, a dutch number as well as my girlfriend's number.
[18] Since TransIP (and also other) require at least two different IP Addresses when configuring the glue records.

- Ubuntu Server 22.04 LTT (Debian)
- IPv4: 194.233.170.54
- IPv6: 2a01:7e01::f03c:93ff:fe27:6b0d
- Bind: 9.18.1-1ubuntu1.2-Ubuntu

A.2 Appendix A.2 - The dummy website

The dummy website is hosted by the Django-python web frame, as it is what I was most familiar with, and was contained in a docker container, which is running on my server. In the end, the website looked as follows:

Fig. 4. Screenshot dummy website at rkreuger.nl.

To be able to get a preview at all of the messenger's chats, multiple attributes needed to be defined in the header of the webpage. The following code is responsible for the example in Figure 5. `<link rel="icon" type="image/png" href="https://i.imgur.com/S41GDYA.png">`

```
<title>Robin Kreuger</title>
<meta property="og:title" content=" Robin Kreuger - Photographer based in Enschede, NL">
<meta name="description" content="Hello, I am Robin Kreuger a student at the University of Twente, and very passioned about photography, programming, and electronics">
<meta property="og:description" content=" Hello, I am Robin Kreuger a student at the University of Twente">
<meta property="og:image" content="https://rkreuger.nl/static/images/ProfilePicturePreviewFacebookNew.png">
```

Fig. 5. Example preview card (Slack).

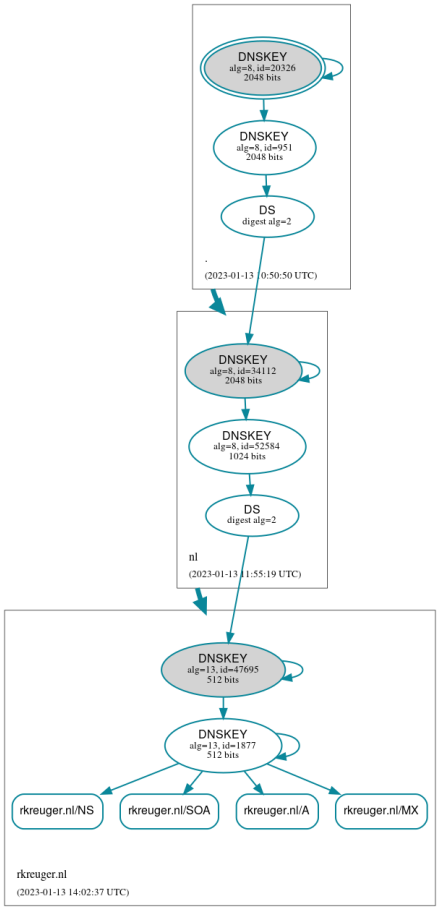

Fig. 6. Correct DNSViz analysis of rkreuger.nl.

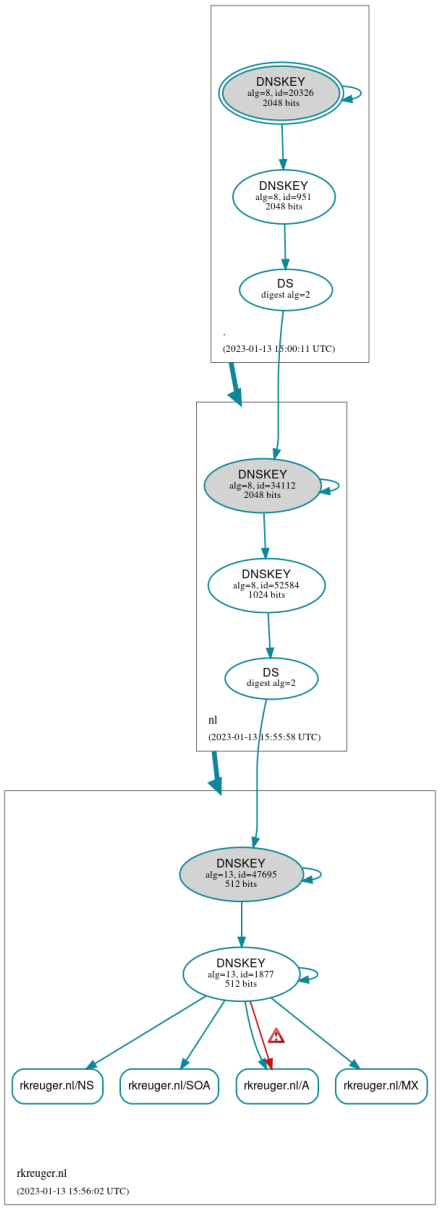

Fig. 7. Bogus DNSViz analysis of rkreuger.nl.