THE THREAT OF STATE-LEVEL SURVEILLANCE

Using HTTPS Interception

Alexandra Dirksen

March 2025

Table of Contents

1. INTRODUCTION **4**

2. MOTIVATION **5**

3. A PRIMER ON ENCRYPTED WEB COMMUNICATION **6**

3.1 The Matter of Trust 7

4. THE CASE OF RUSSIA **8**

4.1 Russia's Path Towards Encapsulation 8

4.2 Russia's Internet Infrastructure 9

4.2.1 Control of the Sub-Network 9

4.2.2 Control of a Certificate Authority 10

4.2.3 Control of a Browser 10

5. INVESTIGATION GOAL AND METHODS **11**

5.1 Scope of this Project 12

5.2 Experiment Infrastructure 13

5.2.1 Crawling Software 13

5.2.2 Domain List 14

5.2.3 List Pre-Processing 15

5.2.4 Data Collection 16

5.2.5 Post-Processing and Analysis 16

6. FIRST CRAWLING RESULTS **17**

6.1 Data Overview 17

6.1.1 Metadata 17

6.1.2 Statistics 18

6.1.3 Russian Trusted Certificate Authority 20

6.2 Deviations 20

6.2.1 Soundness of Deviations 21

6.2.2 Anomaly Concerning RTCA 22

7. INTERPRETATION OF THE RESULTS **24**
7.1 Interpretation of Anomaly 24
7.1.1 Browser-Dependent Trust Policies 24
7.1.2 Possible HTTPS Interception Attempts? 24
7.2 Potential Implications for Users 25
8. CONCLUSION **26**
8.1 Lessons Learned 26
8.2 Presentation of this Work 27
9. ONGOING AND FUTURE WORK **27**
9.1 Anomaly Analysis 27
9.2 Global Scale 28
9.3 Detention of HTTPS Interception 28
10. ACKNOWLEDGEMENTS **29**

1. Introduction

This research examines the potential presence of HTTPS[1] Interception as an attack class for state-level surveillance, focusing on the case of Russia and its infrastructure. What makes HTTPS Interception particularly threatening is the opportunity it holds for states to carry out surveillance on a user without them being aware that their online activity and personal data is under surveillance. This study reveals how Russia's successful efforts in centralizing its digital infrastructure creates the conditions for HTTPS Interception, a reality that raises key privacy concerns for users.

Digital surveillance is an attack that is not obvious: it is difficult to detect, often requires more resources and control by the attacker to execute, and is therefore given less attention in research. Large-scale attacks by state actors that are more visible on the other hand, such as censorship or internet shutdowns, have been widely studied and reported. In these cases, the effects of the attacks are immediately apparent to those affected, whether it be the denial of a single service or website, or restricted or no access to the Internet as a whole.

This research aims to contribute to research on state-level digital surveillance as a field that requires more attention. In examining the potential presence of HTTPS Interception by the Russian state, and given our own resource and time limitations, we paid specific attention to Russia's deployment of the Russian Trusted Certificate Authority (RTCA), the development of which was accelerated due to recent international sanctions.

Through this entry point, we demonstrate in this study how Russia's centralized digital infrastructure, including the state-controlled Yandex browser, enables selective trust in RTCA certificates, which is a precondition for HTTPS Interception. Key findings reveal anomalies in HTTPS certificate chains, with discrepancies depending on browser trust policies, which suggests potential interception or misconfigurations. The report draws attention to the potential risks for users under state surveillance and calls for expanded global monitoring and tools to detect such practices.

After introducing the concept and rationale for encrypted web communication at the start of this report, we present our investigation approach, a novel strategy for detecting HTTPS Interception. Here, we delve specifically into the case of Russia, and the Russian government's efforts for digital control. We then share our initial results and our interpretation of the data. After identifying our lessons learned and the possibilities for expanding the scope of this research globally, we conclude with an outlook on future work.

This project was part of the Information Controls Fellowship Program (ICFP) of Open Technology Fund (OTF), and advised by Censored Planet.

1 Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website.

2. Motivation

Over time, with the expansion of the use of the internet, numerous mechanisms have been developed to protect users' digital privacy. Researchers and privacy advocates calling for the implementation of these mechanisms are often dependent on reports from affected users whose digital privacy has been breached.

However, malicious actors are continuously adapting their tactics in response to these developments. This results in a game of cat and mouse, whereby attackers continually attempt to circumvent or compromise security mechanisms, which in turn are forced to evolve. This is no different for an attacker such as the state. However, due to the state's major influence on their country's digital infrastructure, as well as their resources, they maintain a strong upper hand. Creating the conditions to carry out an attack without the awareness of the user is a significant advantage for the state, and is part of what makes HTTPS Interception so threatening.

The Web PKI[2] is one such playground in which attackers are continuously adapting their tactics. The Web PKI is the primary framework that ensures secure communication over the internet using several components designed to work in tandem. If a malicious actor acquires control of one of the components, they can impair or even overcome this protection mechanism and jeopardize a user's privacy. On the one hand, the successful execution of this attack is not easy. It requires the control of additional components outside the Web PKI, which is possible for an attacker with extensive access to resources, power or expertise, such as a state. At the same time, if successful, this attack is challenging to detect, and protecting users against it is even more challenging.[3]

After Russia invaded Ukraine on 24 February 2022, they were subjected to numerous international political and economic sanctions. As part of the sanctions, western Certificate Authorities (CA)[4] temporarily stopped offering their services to Russia-based top-level domains, resulting in limitations in Russia's digital infrastructure. Consequently, Russia developed its own domestic CA, which is a contradiction: a CA in government hands leads to concerns about political influence and misuse of power, and brings into question their ability to maintain neutrality to protect users' privacy.

Russia's development of a domestic CA again brought to light Russia's attempts to encapsulate its digital infrastructure from the global network. Russia claims this endeavor is for the purposes of putting parts of its digital infrastructure back into national hands, so that it is less dependent on Western companies. The consequence, however, is that these parts become controlled by government powers, making it easier for the state to impose censorship and other digital attacks. One such potential attack is mass surveillance, within the digital space which utilizes a technique called HTTPS Interception. In the simplest case, this man-in-the-middle technique allows the attacker to intercept a user's encrypted internet traffic and read its content. In the worst case, the attacker can forge the content of the data traffic or even direct the user to a fake website. This work investigates the presence of such an attack within Russia's network.

2 Public key infrastructure (PKI) refers to tools used to create and manage public keys for encryption, which is a common method of securing data transfers on the internet.

3 The term for such an attack scenario is called, 'Low Probability, High Impact', introduced by Bussière and Fratzscher. See Bussière, M. and Fratzscher, M. (2008) Low probability, high impact: Policy making and extreme events. *Journal of Policy Modeling 30 (1)*. 111-121.

4 A Certificate Authority is a company or organization that validates the authenticity and trustworthiness of a website, domain or organization so users know exactly who they're communicating with online and whether that entity can be trusted with their data.

3. A Primer on Encrypted Web Communication

Encrypted web communication is fundamental to modern internet security, ensuring that data transmitted between users and websites remains confidential and protected from eavesdroppers. This is typically achieved through protocols such as HTTPS, which encrypts data using TLS.[5] HTTPS aims to provide a private, encrypted 'tunnel', in which the data between two (or more) parties, for example, an internet user and a website's server, can only be read by those two parties.

HTTPS utilizes digital certificates such as one from Figure 1 below:

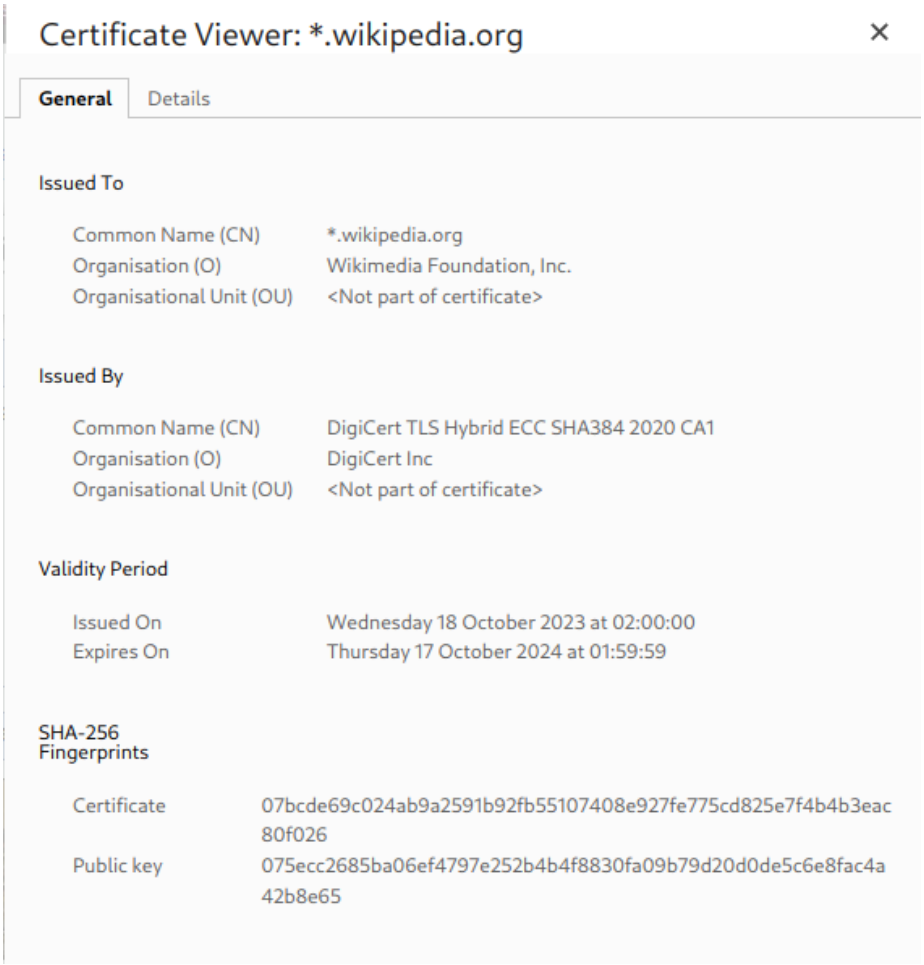

Figure 1: Detail of a certificate for the domain wikipedia.org, viewed in Google Chrome. It shows the domains for which it is valid (wikipedia.org and all subdomains), who the domain owner is (Wikimedia Foundation Inc.), who issued the certificate (DigiCert), and how long the certification is valid (one year from Oct. 18, 2023).

5 Transport Layer Security (TLS) is a cryptographic protocol designed to enable secure communication over the internet.

A digital certificate is provided to a user's browser client when a domain is requested, which is then checked by the client before the encrypted communication is initialized. Secured by cryptography, the certificate confirms the legitimacy of the recipient's ownership and access to a domain in an unforgeable way. In other words, it guarantees the user that they are actually talking to the legitimate owner of the domain in question. Such certificates are issued by an allegedly independent organization, called a Certificate Authority (CA), of the domain owner's choice. The restriction here is that the end-user's browsers must trust the issuing CA when requesting the domain in question in order for the user to access the domain.

Figure 2 shows a simplified initialization of an HTTPS connection:

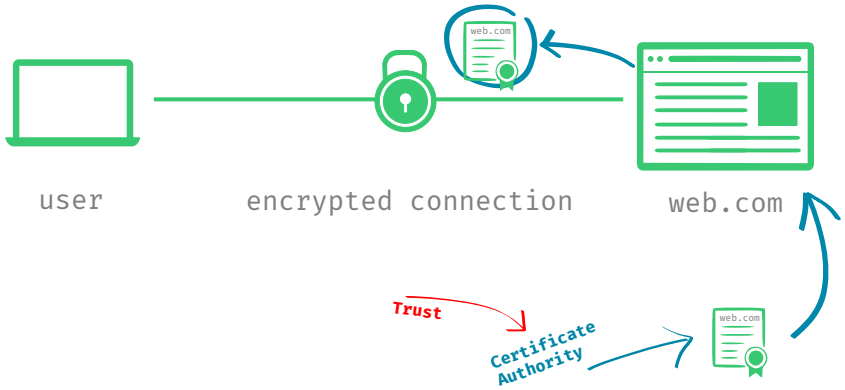

Figure 2: A user's client retrieves a domain's certificate at the requesting attempt.

However, the crux of the matter is this: a common internet user has no choice, but to *trust* its browser client when connecting to a domain. The browser client, in turn, trusts the CAs to do the ownership check properly.

3.1 The Matter of Trust

As indicated above, trust plays a vital role when it comes to security on the web. Additional measures exist to reduce the trust chain, such as Certificate Transparency[6] and Transparency Gossip,[7] for example. However, while the latter has been in the draft stage for years and will probably remain there, we have already proven[8] that the first procedure is not effective in detecting untrustworthy CAs, and therefore has already been excluded from the model.

A CA is rendered *untrustworthy* under two scenarios, among others. For one, this can happen when it omits the domain ownership check and issues a certificate for a specific

6 Certificate Transparency (CT) is an Internet security standard for monitoring and auditing the issuance of digital certificates. See Laurie, B., Langley, A. and Kasper, E. (2013) Certificate Transparency. RFC 6962. *RFC Editor, June 2013*. Available here: https://datatracker.ietf.org/doc/html/rfc6962.

7 Certificate Transparency Gossip is a mechanism where clients and servers share log information to detect misissued or rogue TLS certificates. See Nordburg, L., Gillmor, D. and Ritter, T. (2020) *Gossiping in CT, IETF Draft*. Available here: https://datatracker.ietf.org/doc/draft-ietf-trans-gossip/

8 See previous research: Dirksen, A., Klein, D., Michael, R., Stehr, T., Rieck, K. and Johns, M. (2021) LogPicker: Strengthening Certificate Transparency against covert adversaries. *Proceedings on privacy enhancing technologies (4)*. 184-202. Available here: https://petsymposium.org/popets/2021/popets-2021-0066.php.

domain to someone without ensuring they are the legitimate owner of the domain. While this may happen due to technical errors or security vulnerabilities, this is unlikely. Given the significance of their role on the web, CAs are high-value targets for attack, and thus, it can be assumed that they use reasonable security practices. A second scenario is when a CA omits the ownership check due to the pressure of a malicious controller. This attack is called *Compelled Certificate Creation*, and was introduced by Soghoian and Stamm (2010).[9] Forcing a CA to carry out this attack requires a powerful actor like a state government. When a CA issues a certificate for a domain to someone who is not the legitimate domain owner, a *rogue certificate* is the result, which can be misused to deceive a user's client and intercept the encrypted communication. This work investigated the presence of this exact attack motivated by governments of repressive states.

4. The Case of Russia

Due to the extensive scope of the project and ongoing changes to the political context, we focused our investigation on Russia as a case study. However, it must be said that this scenario does not apply exclusively to Russia. To carry out HTTPS Interception, a state requires control of certain components of their internet infrastructure. Russia was an ideal focus for our investigation, given the measures the Russian government has taken to achieve a "sovereign internet", that have enabled it to obtain the necessary control to carry out such an attack.

4.1 Russia's Path Towards Encapsulation

Russia's efforts to cut itself off from the global network and obtain control of its "sovereign internet" have long been known. Numerous attempts to control digital activity by the Russian government during the last decade have been identified and reported.[10],[11],[12] In recent years, these attempts have included information flow control regarding the COVID-19 pandemic using censorship,[13] and banning specific communication tools, such as Telegram Messenger.[14] Russia's invasion of Ukraine led to further bans of other communication tools, including Meta products (previously Facebook). These bans were justified by the state through Russia's *Yarovaya Law* (2016),[15] which claimed to prevent

9 Soghoian, C. and Stamm, S. (2010) Certified Lies: Detecting and defeating government interception attacks against SSL. Available here: https://petsymposium.org/2010/papers/hotpets10-Soghoian.pdf.

10 See for example, Human Rights Watch (2020) Russia: Growing Internet Isolation, Control, Censorship. *Human Rights Watch*. Available here: https://www.hrw.org/news/2020/06/18/russia-growing-internet-isolation-control-censorship.

11 See for example, Reuters (2021) Russia disconnects from internet in tests as it bolsters security – RBC Daily. *Reuters*. Available here: https://www.reuters.com/technology/russia-disconnected-global-internet-tests-rbc-daily-2021-07-22/

12 See for example, Human Rights Watch (2014) Russia: Halt orders to block online media. *Human Rights Watch*. Available here: https://www.hrw.org/news/2014/03/23/russia-halt-orders-block-online-media.

13 Российскай Федерации. "требований в Роскомнадзор о блокировке недостоверной информации о коронавирусе" (June 2020).

14 Александр Рюмин. "Роскомнадзор начал процедуру блокировки Telegram". *ТАСС* (April 2018).

15 The Government of Russia (2016) "Федеральный закон от 06.07.2016 г. No 374-ФЗ".

"extremism".[16],[17] Further regulations followed, including introducing fines on anonymizers,[18],[19] such as VPN providers, when violating the government's bans.

In November 2019, Russia's government introduced a new regulation called the Sovereign Internet Law,[20] which brought forward a new level of digital control to ensure the success of ban regulations and "protect the internet within Russia from external threats".[21] By this law, internet service providers (ISPs) must allow authorities to reroute internet traffic directly, by installing a deep package inspection hardware distributed to them by the government, called TSPU.[22],[23]

Russia's consistent trajectory of implementing measures that give the state ever-increasing control of the digital infrastructure within its borders is of high concern. It holds significant consequences for the rights of people within Russia to access the internet and information freely, without censorship or surveillance.[24]

4.2 Russia's Internet Infrastructure

To carry out HTTPS Interception, Russia, the 'attacker' in our case, needs the control of at least three vital components of their internet infrastructure: the sub-network, a Certificate Authority, and a browser. Russia has obtained control of all three to carry out such an attack successfully.

4.2.1 Control of the Sub-network

Since the enactment of the *Sovereign Internet Law*, Russia obligated all ISPs in the country to install home-grown devices for deep package inspection, called *TSPU*. These devices have allowed the government to control or reroute traffic in a centralized manner.[25] The attacker is thus in control of the first component: the sub-network within their frontiers.

16 Epifanova, A. and Dietrich, P. (2022) Russia's Quest for Digital Sovereignty: Ambitions, realities, and its place in the world (DGAP Analysis, 1). Forschungsinstitut der Deutschen Gesellschaft für Auswärtige Politik e.V. Available here: https://nbn-resolving.org/urn:nbn:de:0168-ssoar-77994-6

17 See also, Sauer, P. (2022) Russia bans Facebook and Instagram under 'extremism' law. The Guardian. Available here: https://www.theguardian.com/world/2022/mar/21/russia-bans-facebook-and-instagram-under-extremism-law.

18 The Government of Russia (2018) Federal Law No. 155-FZ.

19 An anonymizer is an instrument with which a user can change their IP address, and, in doing so, access a censored website from another country where that website is accessible, while staying undetected. Virtual Private Networks (VPNs) use encryption to extend a private network over a public network, such as the internet.

20 See footnote 18.

21 The law and its consequences have been analyzed in detail by Alena Epifanova. See, for example, Epifanova, A. (2020) Deciphering Russia's Sovereign internet law: Tightening control and accelerating the Splinternet. Forschungsinstitut der Deutschen Gesellschaft für Auswärtige Politik e.V. Available here: https://www.ssoar.info/ssoar/handle/document/66221.

22 технические средства противодействия угрозам ("Technical measures to combat threats"), known colloquially as TSPU.

23 Xue, D., Mixon-Baca, B., ValdikSS, Ablove, A., Kujath, B., Crandall, J. and Ensafi, R. (2022) TSPU: Russia's Decentralized Censorship System. ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France. ACM, New York, NY, USA. Available here: https://censoredplanet.org/assets/tspu-imc22.pdf.

24 To shed light on Russia's long-term goals, Epifanova and Dietrich (see footnote 16) explored Russia's concept of digital sovereignty, their vision of a smart economy, and the domestic legitimization of those plans through laws such as the Sovereign Internet Law.

25 Xue et. al. have investigated the presence of those devices in the Russian network in detail. See footnote 23.

4.2.2 Control of a Certificate Authority

Shortly after being subjected to sanctions in 2022, Russia introduced a domestic CA called “Russian Trusted Certificate Authority” (RTCA).[26],[27] By misusing their control, the attacker could compel this CA to omit the domain ownership check and issue certificates for domains they do not own but wish to intercept. The control of a CA forms the second component needed to perform the attack.

4.2.3 Control of a Browser

Controlling a CA is of no use when it is not trusted by browsers, since its issued certificates would not be accepted. The launch of RTCA immediately raised questions for browser vendors about whether browsers should trust root certificates issued by the RTCA or not.[28] None of the most commonly-used browsers, such as Chrome, Firefox or Safari, currently support it. The Russian-Dutch browser, Yandex, thus plays a vital role in Russia’s digital landscape, since it supports the Russian government’s decisions. Yandex includes the RTCA in its trusted root store.

Figure 3 shows the results of requesting the same website using Chrome, Firefox, and Yandex. The website in the illustration is protected using an RTCA-signed certificate:

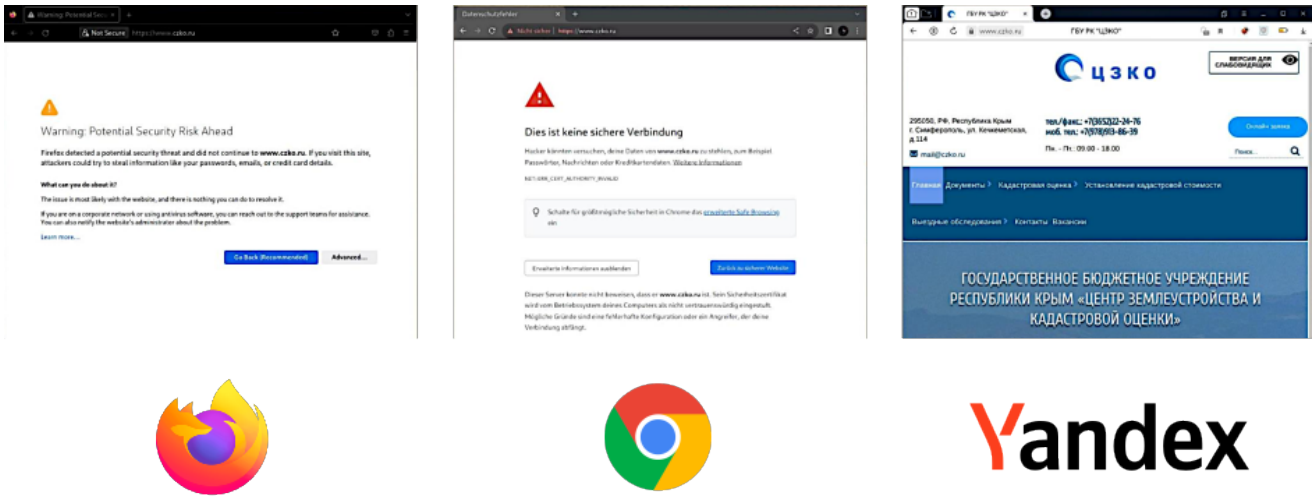

Figure 3: Requesting the same domain from three browsers. The domain’s web server is serving a certificate signed by RTCA. Firefox and Google Chrome do not trust RTCA and do not forward the request. Yandex, however, completes the request.

26 Получите электронный сертификат безопасности (2022) Available here: https://www.gosuslugi.ru/tls. Date accessed: 4 April 2024.

27 Shahzad, I. (2022) Russia establishes its own TLS Certificate Authority to avoid sanctions. *Medium*. Available here: https://medium.com/coinmonks/russia-establishes-its-own-tls-certificate-authority-to-avoid-sanctions-a8221b72b729.

28 See, Russia preparing for MitM. Mar. 2022. Available here: https://groups.google.com/a/mozilla.org/g/dev-security-policy/c/QaKxfr5hOXg.

From a global view, Yandex's market share is negligible at less than 1%. But in Russia, its market share is constantly growing, from 19% in August 2023, to over 22% at the time of writing.[29] In February 2024, the former Dutch-based owner of Yandex sold the browser entirely to a Russian consortium of investors.[30] Since then, Russia can be assumed to be in control of a domestic web browser that trusts its domestic CA. This is the last of the three components to perform the introduced attack.

5. Investigation Goal and Methods

From a technical view, the goal of this research was to detect possible ongoing HTTPS Interception attacks, and contribute to an infrastructure for future long-term monitoring.

When a website is requested by a browser client via HTTPS, the client and the requested web server exchange handshake data. This data is needed to establish an HTTPS connection, and contains the web server's certificate, among other information.

To achieve our goal, our strategy was to request the exact same domains from as many geolocations as possible to obtain handshake data for this request. The optimistic assumption was that the handshake data, particularly the web server certificate, would remain the same for each geolocation and domain accordingly. Should the handshake data differ, we would have to find a meaningful technical explanation for this. If no meaningful technical explanation could be found, those cases must be considered anomalies, which we would check manually in greater depth for signs of HTTPS Interception attacks.

Figure 4 shows a very simplified presentation of the intended crawling procedure:

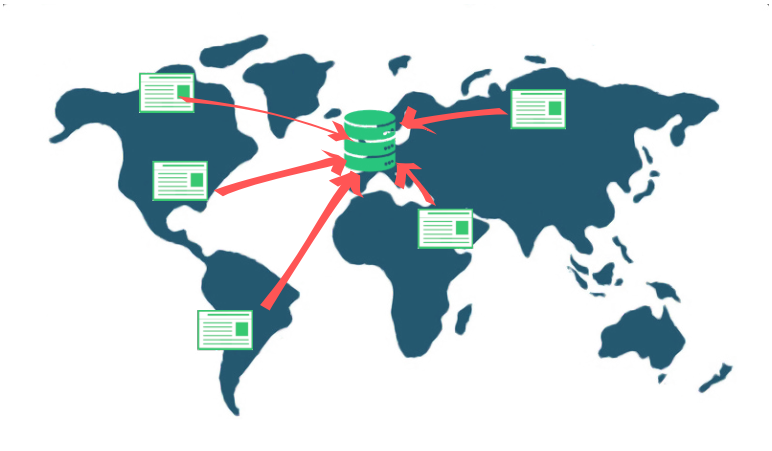

Figure 4: Simplified presentation of the crawling procedure, where one web server is requested from five clients from different geolocations.

29 Statcounter (2025) *Browser Market Share Russian Federation*. Available here: https://gs.statcounter.com/browser-market-share/all/russian-federation.

30 Mariko Oi (2024) Yandex: Owner of 'Russia's Google' pulls out of home country. *Kommersant*. Available here: https://web.archive.org/web/20220306183205/https://www.kommersant.ru/doc/5249500.

As such, we collected handshake data to search for rogue certificates. Recognizing a rogue certificate is not an easy task. When data is investigated from a malicious vantage point (VP), it almost always appears valid. Detection of malicious data also requires the researcher to have trust in their infrastructure, and in other words, the certainty that they themselves are not operating under a malicious VP. This could not be ensured in our case. However, since we assumed that the attacker's power was limited to its frontiers, we set about collecting and comparing data from as many VPs as possible, based outside of the attacker's frontiers. In this way, we could obtain a homogeneous dataset, where data from malicious VPs stands out as an anomaly and can then be reviewed manually.

5.1 Scope of this Project

Due to time and resource limitations, as well as continuous changes to the Russian political context, we focused our data collection specifically on one VP in Russia, and one VP in Germany, which served as the control.

For our attack model, we assumed the attacker would act like a 'covert adversary,' a model introduced by Aumann and Lindell in 2010.[31] Such an attacker is only willing to perform an attack if they are not caught, both during and after the act. We believed this adversary aligns with the digital and political context of the present case in Russia: the government, as an attacker, is willing to attack their citizen, or in other words, intercept their internet connection. However, they would want to avoid being caught, since this could result in a loss of reputation and trust among the state's citizens. This assumption then led to our next assumption, that the attacker only targets VPs that behave like a human internet user, to avoid being detected by a monitoring infrastructure built, for example, by internet activists, or for research, and risk getting exposed.

We built a crawling infrastructure capable of collecting handshake data, specifically certificates, for a manually-curated[32] list of domains from VPs in Russia and Germany. The software prototype was built in a way that it could be adapted or included in monitoring projects that were already operating, such as censorship research projects OONI,[33] The Citizen Lab,[34] or Censored Planet.[35] We were aware that our dataset could potentially reveal further insights into the deployment of TLS in Russia and Germany, as well as other security issues we may not anticipate at the start of this project. We therefore collected the whole TLS handshake in our data collection process, not only the certificates, in order to support the possibility of future investigation.

Finally, for the purposes of future work, we make a proposal in this report for a user-friendly detection mechanism that could be used directly by internet users. Due to the characteristics of our research, protection of internet users was not directly within this project's scope, but remains relevant for further exploration.

31 Aumann, Y. and Lindell, Y. (2010) Security against covert adversaries: Efficient protocols for realistic adversaries. Journal of Cryptology 23 (2). 281–343.

32 Manually-curated means here that we removed all domains known to be subject to Russian censorship.

33 *Open Observatory of Network Interference*. Available here: https://ooni.org/de/. Date accessed: 10 June 2024.

34 *The Citizen Lab*. Available here: https://citizenlab.ca/. Date accessed: 10 June 2024.

35 *Censored Planet*. Available here: https://censoredplanet.org/. Date accessed: 10 June 2024.

5.2 Experiment Infrastructure

We built the infrastructure with a number of considerations. First, it needed to include an automated web browser and be configured to simulate a typical internet user's activity from those VPs as much as possible. Second, in order to map the issues articulated above regarding the RTCA, the domain requests needed to be made from at least two browsers, Yandex and, for example, Chrome.

Usually, if a browser is served an untrustworthy certificate,[36] it declines the connection and/or responds with a TLS error, as shown in Figure 7 (see page 24). Our browsers are configured to ignore TLS errors, and also accept certificates that they do not trust. It is necessary to obtain any certificates served and examine their content before they are discarded.

Lastly, given that crawling a high number of domains in parallel produces a high load and requires time, the software ran on a server equipped with AMD EPYC 7713 with 2x32 cores and 126 GB memory. In this way, the experiments could be controlled from any terminal with VPN access.

5.2.1 Crawling Software

Since both browsers used for this study, Chrome and Yandex, use Chrome as a back-end,[37] we used Puppeteer[38] as the base for the crawler software. Puppeteer is a node.js library that allowed us to instrumentalize Chrome via Chrome's DevTools protocol. Puppeteer provides the headless[39] parameter to prevent the browser window from spawning.[40] The software is controllable only via a programmable API,[41] which is vital for browser automation. In this way, we could customize the browser according to our needs, request a domain using different parameters, intercept the connection, and collect the results, which in this case, was the handshake data.

However, requests using an automated headless browser are detectable by web servers. Those domain owners who don't want their domains to be on crawling lists tend to block the requests. To overcome this issue, we ran the crawler in headful mode and redirected all graphical outputs to a virtual frame buffer using the tool Xvfb.[42],[43] We used a Puppeteer Cluster[44] to speed up the crawling procedure, by using its feature to create multiple puppeteer workers, which are automated concurrently.

To be able to retrace the data collection afterward, it was essential to know what the

37 When a browser uses Chrome as a back-end, it means that the core functionality of the browser, including the rendering engine and underlying technology, is based on the same codebase as Google Chrome.

38 *Puppeteer*. Available here https://pptr.dev/. Date accessed: 10 June 2024.

39 In the newest version of Chrome, the headless flag has been deprecated. However, the current crawling software depends on an older version for technical reasons and will be updated in the future.

40 Automated crawling of a large list of domains (for example, a list of 10,000 domains) requires the browser window to remain invisible. Otherwise, every domain would be opened in its own browser window, rendering the procedure infeasible in terms of performance, since browsers require a high amount of system resources.

41 An application programming interface (API) is a connection between computers or between computer programs.

42 xvfb - *Linux man page*. Available here: https://linux.die.net/man/1/xvfb. Date accessed: 10 June 2024.

43 For this, we allowed the browser to render it's graphical window, which is not shown on a physical monitor. Instead, it is redirected to *a virtual frame buffer*, which allows us to render it invisible (see footnote 40).

44 *Puppeteer Cluster*. Available here: https://github.com/thomasdondorf/puppeteer-cluster. Date accessed: 10 June 2024.

network data looked like during the handshake initialization. In other words, we needed to know which parameters were sent during the domain requests from each browser, and precisely what the response looked like. For this reason, we also monitored our network interface during the entire crawling procedure, and recorded the data using the tool tcpdump.[45] This would also be helpful during anomaly analysis, when results for the same domain differ in both browsers.

For data storage, we used a PostgreSQL database.[46]

5.2.2 Domain Lists

To perform HTTPS Interception, Russia as the attacker must employ a certificate that RTCA issued. However, as shown by Jonker et al.,[47] the distribution of RTCA-signed certificates is scarce compared to the usual global top domain lists used for crawling, such as the ones provided by the Tranco Project.[48] The domain lists they provide mainly aggregate frequently used websites from a global view, collected from different sources. To increase the chance of detecting domains protected by RTCA, we needed a domain list that primarily targets domains relevant to Russian users. Since the only localized top-domain collection has been deprecated for a long time,[49] we needed to create our own list. In order to incorporate as many Russia-related domains into this study as possible, our final domain list was a merged collection from different sources. These sources included:

Alexa RU
Even if Alexa's localized lists are deprecated, we included the last known top-1,000 domains for Russia, as they are still accessible.

RTCA list
The website of RTCA lists domains for which they have issued certificates.[50] It contains 4,883 domains, including 2,854 wildcards.

CloudFlare Radar
We obtained a manually curated top-1,000 domain list for Russia directly from CloudFlare's Radar project.[51]

Tranco (customized)
Tranco offers research-oriented top-sites ranking lists that can be customized for premium customers.[52] Our custom list consists of 228,329 domains with top-level domain .ru collected from different projects, including Umbrella, Majestic, and Crux.

45 *TCPDump* and *Libpcap*. Available here: https://www.tcpdump.org/. Date accessed: 10 June 2024.

46 While the technical details regarding the crawling software go beyond what has been described, they can be obtained in the GitHub project or directly requested if needed.

47 Jonker, M., Akiwate, G., Affinito, A., Claffy, k., Botta, A., Voelker, G., van Rijswijk-Deij, R. and Savage, S. (2022) Where .ru?: Assessing the impact of conflict on Russian domain infrastructure. *IMC '22: Proceedings of the 22nd ACM Internet Measurement Conference*. 159-165.

48 *Tranco: A research-oriented top-down sites ranking hardened against manipulation*. Available here: https://tranco-list.eu/. Date accessed: 10 June 2024.

49 *About Alexa Internet* (archived). Available here: https://web.archive.org/web/20091007102542/https://www.alexa.com/company. Date accessed: 10 June 2024.

50 *Gosuslugi*: List of domains signed by Russia's CA. Available here: https://www.gosuslugi.ru/api/nsi/v1/custom/dic/tls/csv. Date accessed: 15 May 2022.

51 *CloudFlare Radar*: Domain Rankings. Available here: https://radar.cloudflare.com/domains. Date accessed: 10 June 2024.

52 See footnote 48.

CT-sans
CT-sans is a tool that automatically downloads certificates from CT-logs[53] and extracts all Subject Alternative Names (SAN) from them. We used all domains with top-level domain .ru from one CT-sans output. This resulted in 13,081,666 domains, including 895,114 wildcards.[54]

DB-IP Probing
This method is the most potent but also time-consuming. The project DB-IP[55] collects IP ranges from all worldwide announced IPs. We first extracted all IP prefixes from ISPs located in Russia (187,068) and probed each resulting IP address at port 443, which is the default HTTPS port. Where the request was successful, we obtained each certificate's SAN for our list. This method resulted in 42,267 domains, including wildcards.

5.2.3 List Pre-Processing

Our domain collection contained over three million domains that needed to be pre-processed before starting the crawling procedure.

Wildcards
Our collection contained a lot of wildcard domains (*.domain.tld), which must be resolved to whole domains to be usable for crawling. We used the tool Findomains[56] to resolve each wildcard into all valid domains known by common DLS resolvers and added them to our collection.

Reachability
To speed up the final crawling procedure, we probed each domain on our list in advance to ensure reachability. Each domain that did not respond was removed from the list.

Broken Subdomains
We discovered a high number of domains containing multiple aggregated .git subdomains in the form of '.git.git.git.git....domain.ru'. After further investigation, it seemed that each affected domain used the same Russian website-building framework named Tilda,[57] and included a certificate from the company Let's Encrypt.[58] This is not problematic, per se, since the certificate renewal of Let's Encrypt is usually automated. However, some domains included more than ten .git subdomains. A possible explanation is that Tilda uses a broken script for their certificate renewal, which produces certificates by adding a new subdomain instead of exchanging the old ones. Therefore, we removed each domain from our list that contained more than one .git subdomain.

53 Certificate Transparency (CT) logs are public records that track the issuance of SSL/TLS certificates. They are designed to be auditable, so anyone can verify that a certificate is legitimate.

54 A wildcard domain is a domain name that uses an asterisk (*) as the first part of its name, allowing it to match any subdomain under that main domain (for example, a certificate for the wildcard domain *.*opentech.fund* also applies to sub.opentech.fund).

55 DP-IP: IP geolocation API and database. Available here: https://db-ip.com/. Date accessed: 10 June 2024.

56 *Findomain Monitoring Service*. Available here: https://github.com/Findomain/Findomain. Date accessed: 10 June 2024.

57 *Tilda Publishing*. Available here: https://tilda.cc/tpls/. Date accessed: 10 June 2024.

58 *Let's Encrypt*: A nonprofit Certificate Authority providing TLS certificates. Available here: https://letsencrypt.org/. Date accessed: 10 June 2024.

Censored Domains

As a final preprocessing step, we needed to filter our domain collection for domains censored in Russia. This protects the rightful owner of the VPN endpoint in Russia to which we would be granted access. Those domains would not be accessible from Russia anyway, due to the blocking mechanisms used to implement censorship.

Russia's Federal Service for Supervision of Communications, Information Technology and Mass Media (Roskomnadzor), provides a website[59] for users where they can check whether a domain is included in Russia's blocking list without actually requesting the domain directly. The project Zapret collected a list of all domains reported to be blocked by this website,[60] which we used as a filter for our domain collection.

Finally, we sanitized the list, as some domains were invalid, for example, due to containing invalid characters. After undergoing all preprocessing steps, our domain list contained **1,626,595 domains**, which were then used for the crawling procedure.

5.2.4 Data Collection

The data collection involved two VPs, one from a research institution based in Germany, and one from a commercial Russian VPN provider, located within Russia. Since the use of a VPN is visible for network monitors, and VPN use is restricted in Russia, our connection was established using Outline.[61] Each crawl was performed using two browsers, Yandex and Google Chrome, which makes two datasets per VP. We used enhanced methods for VPN hiding (Shadowsocks Protocol). If a request was redirected, we always processed the whole redirection chain until the last step and collected the data from the final step accordingly. Each domain was requested using the HTTPS protocol preamble. Given the limitation of our hardware resources and the network's natural response time, each crawl had a runtime of approximately ten days, with 500 requests in parallel.

5.2.5 Post-Processing and Analysis

To analyze the data after successful collection, we mostly relied on two Python Libraries: Pandas,[62] a library for data analysis, and NumPy, which allows performant scientific computation. We used different Python scripts for processing the collected data, extracting statistics from it, and performing anomaly analysis, after further pre-filtering.

59 This is a register of the domain names, website references and network addresses that contain information that is forbidden in the Russian Federation. Available here: https://eais.rkn.gov.ru/en/. Date accessed: 10 June 2024.

60 *Register of Internet Addresses filtered in Russian Federation*. Available here: https://github.com/zapret-info/z-i. Date accessed: 10 June 2024.

61 *Outline*. Available here: https://getoutline.org/. Date accessed: 24 February 2025. Outline is an open source tool developed by Google Jigsaw. It is designed to help users in areas with restricted internet access to create their own VPN (Virtual Private Network). The primary purpose of Outline is to allow users to bypass censorship and access the internet securely.

62 *pandas*: Python Data Analysis Library. Available here: https://pandas.pydata.org/. Date accessed: 10 June 2024.

6. First Crawling Results

Since we have only recently finished building the crawling infrastructure, the second and main round of data collection is still running. The anomaly analysis of the data being collected is therefore still ongoing.

6.1 Dataset Overview

The following section presents an overview of the data that emerged from the first crawl, including the metadata and statistics, as well as the statistics related specifically to the RTCA.

6.1.1 Metadata

Table 1 shows a summary of the results from the first round of data collection, which contains four datasets in total. We used the same domain list for each round of data collection, described above in detail (see Section '5.2.2 Domain Lists'). In Table 1, row 1 contains the number of all responses, independent of their outcome. Although the number of requests is equivalent to the length of our domain list and remains the same for all sets, we included it for clarity in row 0. In a perfect scenario, the number of requests and responses would be equal. However, since internet measurements are never entirely deterministic, each request is unlikely to be handled correctly, such that the number of responses is slightly smaller. For the initial study, we decided to tolerate this delta as long as it remained within a discrepancy of less than 2%. The missing domains are marked in a separate list to be revised manually, if needed.

Rows 2-8 splits up the number of responses into different categories. Some are related to official HTTP Status codes, some are not. However, the code and result message are stored in a separate table. Domains that appear unresponsive are retried three times until the result of each request is assigned to one of the following categories:

1. The **valid chain** category is for each domain that eventually responds with a valid certificate chain. Since many domains in our list were found using IP scanning, many of them were not actively being used. Given this fact, the number of valid certificate chains seemed sufficient (67% average in Russia, and 61% average in Germany).

2. If the request ended up being redirected to a non-HTTPS domain, in other words, where no certificates are served, the domain was added to the **no certs** category.

3. If the domain was unreachable, in other words, there was no endpoint listening to requests, the domain was added to **no response**.

4. If a domain name could not be resolved, the request results in a **DNS error**.

5. A **timeout** usually indicates erroneous web server configuration, such as redirection loops or a loss of connection.

6. In some cases, the connection can be **refused** by the web server. This may happen if, for example, geo/domain blocking is active, or the crawler is identified as a

bot. Even though we used methods that helped us hide our VPN connection or bypass bot detection mechanisms, they were insufficient in exceptional cases.

7. If requests result in unknown errors, such as redirection errors, due to missing location headers or similar technical issues, the domain was assigned to **miscellaneous** errors.

1. Result class	2. Yandex RU	3. Chrome RU	4. Yandex DE	5. Chrome DE
0. requests	1,626,595	1,626,595	1,626,595	1,626,595
1. responses	1,598,843	1,601,744	1,606,515	1,607,218
2. valid chains	1,076,544	1,035,797	987,435	948,788
3. no certs	153,373	185,793	212,892	238,744
4. no response	47,574	46,430	15,716	16,277
5. DNS error	154,264	150,685	150,288	149,557
6. timeout	151,420	168,920	225,818	239,389
7. refused	1,100	1,004	1,272	1,200
8. misc.	14,568	13,115	13,094	13,263

Table 1: Overview of each list's metadata

6.1.2 Statistics

Cryptographic Parameter

Table 2 shows the top five cryptographic algorithms used throughout the datasets. Since this statistic remained equal for all four collections, it is summarized in one column.

Rank	Algorithm
1	wSHA 256 with RSA
2	ECDSA with SHA384
3	SHA1 with RSA
4	ECDSA with SHA256
5	SHA 384 With RSA

Table 2: Ranking of the algorithms used among all four datasets

Table 3 shows the top five key sizes[63] for cryptographic keys among the Russian and German datasets. The smaller key sizes (256 and 384) refer to a newer encryption standard (ECDH).

Rank	RU	DE
1	2,048	2,048
2	4,096	4,096
3	256	256
4	384	384
5	1,024	3,072

Table 3: Ranking of the key sizes used among the Russian and German datasets

63 For rows 1, 2 and 5 the recommended key sizes are > 3.000. For rows 3 and 4, the recommended sizes are >250. This is according to the recommendations published by the Federal Office for Information Security, Germany: Federal Office for Information Security (2024) BSI TR-02102-2 "Cryptographic Mechanisms: Recommandations and Key Lengths: Use of Transport Layer Security (TLS)" Version: 2024-1. Available here: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-2.pdf. Date accessed: 24 February 2025.

Table 4 shows the ranking of the certificate validity among all datasets in months. As indicated below, the largest number of domains had a certificate validity of three months, the second largest number of domains had a certificate validity of 13 months, with the smallest number of domains having a certificate validity of 12 months.

Rank	valid (months)
1	3
2	13
3	2
4	120
5	12

Table 4: Ranking of the certificate validity among all datasets in months

Certificate Authorities

Tables 5 and 6 show the top seven CAs and CA vendors. The first table below shows that CAs usually do not directly issue certificates but entrust Intermediate Certificate Authorities (ICAs) to do this. If an ICA gets breached, only the subset of certificates signed by the affected ICA must be revoked. Rows 1 and 6 from Table 5 are ICAs from the CA vendor in row 1 from Table 6. Let's Encrypt is by far the most used CA nowadays. It is free of charge, and renewal can be automated.

Rank	Certificate Authority
1	Let's Encrypt R3
2	AlphaSSL - SHA256 - G4
3	GTS CA 1 P5
4	GlobalSign GCC R3 2020
5	GlobalSign RSA OV 2018
6	Let's Encrypt E1
7	GoDaddy G2

Table 5: Ranking of the top five used CAs among all four datasets

Rank	CA vendors
1	Let's Encrypt
2	AlphaSSL (GlobalSign)
3	Google Trusted Services
4	GlobalSign
5	GoDaddy
6	Sectigo
7	DigiCert

Table 6: Ranking of the top seven CA vendors among all four datasets

6.1.3 Russian Trusted Certificate Authority

Since Russian Trusted Certificate Authority (RTCA) appears far down the list of most used CAs, Table 7 shows the occurrences of RTCA specifically. The fewer occurrences within the datasets from Germany stem from the fact that the number of valid chains retrieved from Germany was also lower. However, the low usage of RTCA reflects the findings of Jonker et al. (2022),[64] even if considering our list to include only domains with *TLD=.ru* in contrast to their domain list.

RU	DE
648 Chrome	**524** Yandex
638 Yandex	**455** Chrome

Table 7: Number of occurrences of RTCA among all four datasets

6.2 Deviations

The following section provides statistics that are relevant for our present study and serves as a foundation for future investigations.

The most interesting category for our analysis is that of deviations. Domain requests can end up in different destinations when requested from different VPs due to a number of reasons, such as the language localization of a website's content. This approach may result in different certificates based on the client's geographical location or browser characteristics. However, if the destination remained the same for both VPs (domain name and IP address), but the certificates differ, we considered this an **anomaly**. It is noticeable that the number of anomalies in Russia's dataset is almost four times higher than in the German dataset.

The **8,309** deviations found within the **Russian** dataset can be divided into three categories as shown in Table 8 below:

CAs differ, same destination	89
one CA empty[65]	7,835
CAs differ, destinations differ	385

Table 8: Categories of deviations within the Russian dataset

The **6,068** deviations found within the **German** dataset can be divided into three categories as shown in Table 9 below:

CAs differ, same destination	55
one CA empty	4,535
CAs differ, destinations differ	1,478

Table 9: Categories of deviations within the German dataset

64 See footnote 47.

65 In this case, only one of both domains provided a certificate. The other domain was not reachable via HTTPS at all.

6.2.1 Soundness of Deviations

Detecting an HTTPS Interception attack is not an easy endeavor, since not each anomaly found can be considered a trace of such an attack. Although rare, different reasons exist for such anomalies. A domain might use different certificates to provide the broadest possible support. Therefore, it is crucial to identify all technically sound reasons for domains to deploy different certificates. In our study, we considered the following:

Compatibility
The client's operating system may support different SSL/TLS protocol versions and cipher suites.

CA Trust
The trusted root stores, or in other words, the list of CAs trusted by a browser, may differ from the trusted root stores of other browsers.

Performance
Depending on the cipher suites used, websites may offer different certificates for mobile clients to enhance performance during the TLS handshake.

Certificate Types
Websites may use different kinds of certificate types, such as EV (Extended Validation), and DV (Domain Validation), for different clients. Depending on the screen size, this can be an option if different websites are utilized for desktop and mobile clients.

Regional Regulations
Some regions have specific requirements for certificates. If the server can detect the browser's locations, it might respond with different certificates to meet those requirements.

The latter case is the one we investigated in this project, since regional regulations are a prerequisite for performing HTTPS interception. While connecting to a web server, the user's browser client sends information, such as its device's operating system, browser type and version, screen resolution, language settings, installed fonts, and plugins. By combining those characteristics, a browser provides a 'unique' fingerprint, allowing the web server to identify the user's client browser. This process is called browser fingerprinting,[66] and is used by web servers for different reasons, such as personalized advertising, analytics, or fraud prevention.

To minimize browser fingerprinting, we took into consideration all deviations from the beginning of the connection request, by investigating the client's TLS packet containing the 'Client-Hello' message. This packet is usually first sent by a client to the web server. By comparing the Client-Hello messages of both browsers for the same domain, we can get an idea of the possible differences, which may lead to fingerprinting. Figure 5 shows an example Client-Hello message sent by the Yandex browser to the domain *19dx.ru*.

66 Zhang, D., Zhang, J., Bu, Y., Chen, B., Sun, C. and Wang, T. (2022) A survey of browser fingerprint research and application. Wireless Communications and Mobile Computing 2022 (1), 3363335.

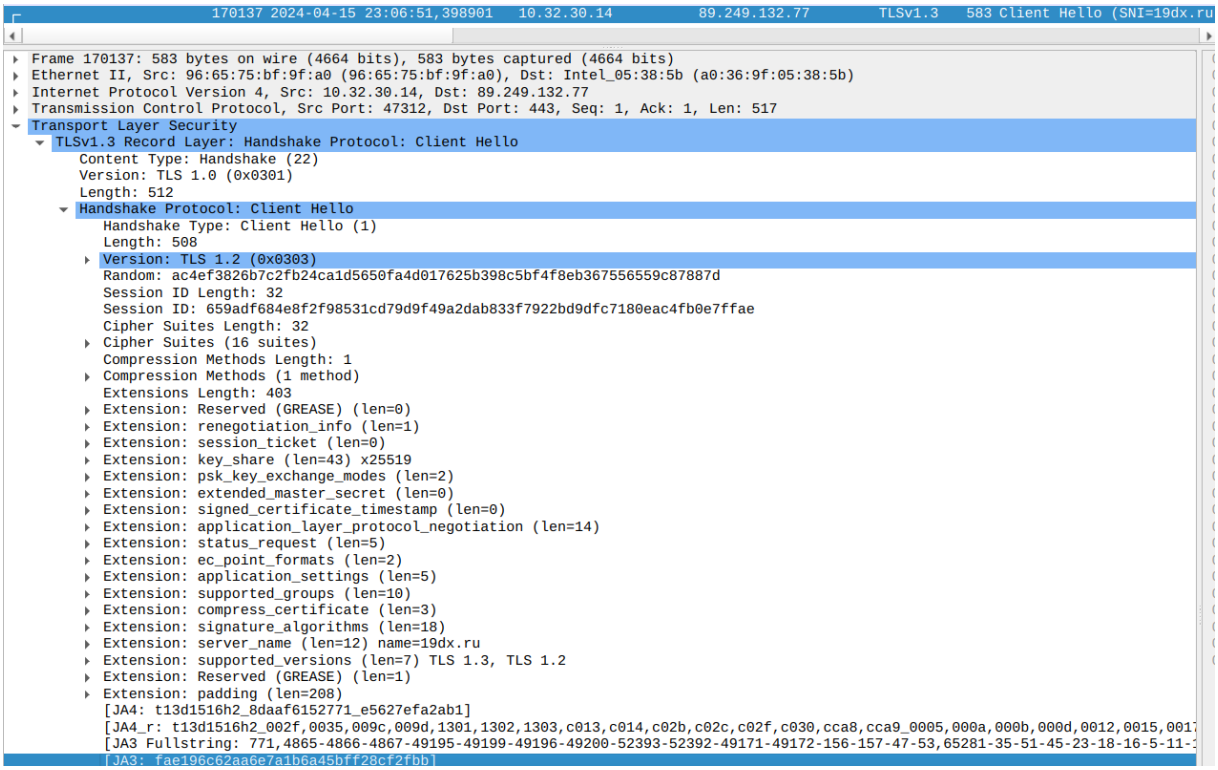

Figure 5: The Client-Hello packet, sent by the Yandex browser to the domain 19dx.ru and captured by Wireshark.[67]

Furthermore, large websites often use several data centers worldwide to facilitate faster TLS handshakes and minimize latency. This process is called load balancing and may lead to each participating data center employing its own certificate. However, detecting load balancing operations is challenging because they are designed to be invisible to the end users. The identification and classification of load balancing is the subject of current research,[68],[69] and part of our future work. (Section 7).

6.2.2 Anomaly Concerning RTCA

During the analysis of our datasets, one interesting anomaly (explained below) emerged in relation to the RTCA. This finding highlights potential vulnerabilities and raises questions about HTTPS Interception within Russia's internet infrastructure. Here, we focused on certificate chains served for the domain *https://rsins.ru* to Chrome and Yandex, when requesting from the Russian vantage point. This domain is hosted by an insurance agency, which belongs to one of the leading consumer brands in Russia, called Russian Standard Corporation. Figure 6 shows a screenshot of the full certificate chains for Yandex and Chrome, and the chain served when the domain is requested using an OpenSSL[70] client.

67 *Wireshark*. Available here: https://www.wireshark.org/. Date accessed: 24 February 2025. Wireshark is a widely used tool for analyzing and visualizing ongoing network communication.

68 Kumar, K. V., Reddylatha, G., Sindhu, M. and Jayasree, K. (2024) A Comprehensive survey of Load Balancing Techniques-From Classic Methods to Modern Algorithms. *International Research Journal on Advanced Engineering Hub (IRJAEH)* 2 (2). 287–296.

69 Almeida, R., Cunha, I., Teixeira, R., Veitch, D. and Diot, C. (2020) Classification of load balancing in the internet. *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. 1987–1996.

70 OpenSSL is a software library for applications that provide secure communications over computer networks to protect against eavesdropping. It is widely used by Internet servers, including the majority of HTTPS websites.

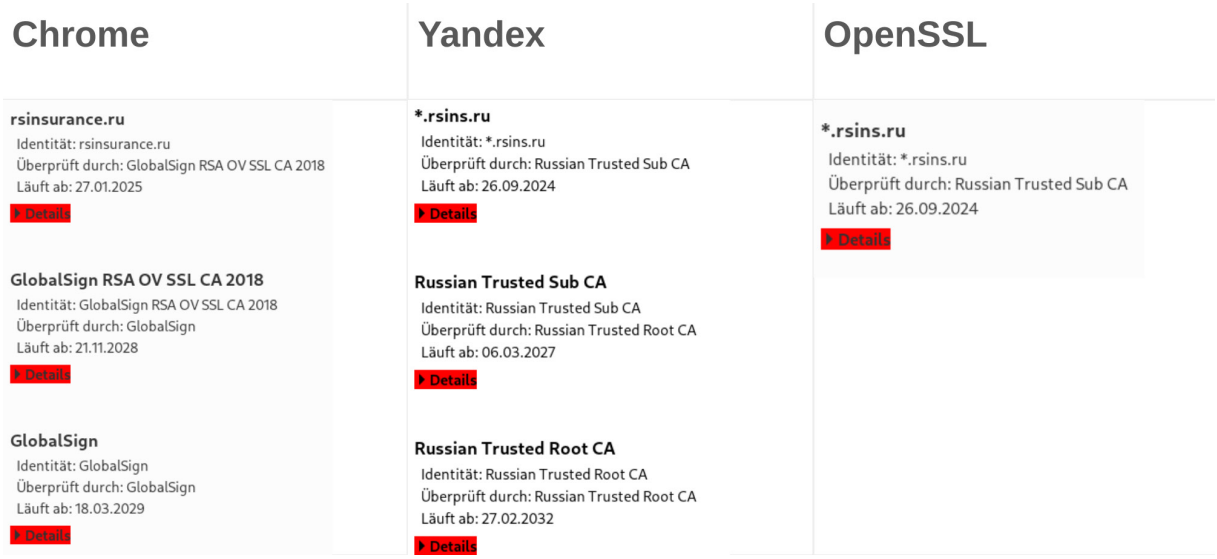

Figure 6: Certificate chains for https://rsins.ru served to Yandex, Chrome and OpenSSL.

Tables 10 and 11 show an extract from the resulting leaf certificates, the last certificates in the chain issued directly for the domain in question. These results indicate that while the domain and IP address remain consistent across requests, the Certificate Authority differs between the browsers.

CA	Russian Trusted Sub CA
validity	12 months
algorithm	SHA256 with RSA
key size	2,048
destination	https://rsins.ru
destination IP	185.71.67.101

Table 10: Last certificate in the chain issued for https://rsins.ru served to the Yandex client

CA	GlobalSign RSA OV SSL CA 2018
validity	13 months
algorithm	SHA256 with RSA
key size	2,048
destination	https://rsins.ru
destination IP	185.71.67.101

Table 11: Last certificate in the chain issued for https://rsins.ru served to the Chrome client.

The differences in certificates served to Yandex and Chrome browsers highlight how each browser's root store policy influences certificate acceptance. Unlike Yandex, which trusts RTCA by default, Chrome does not. As a result, requests from Chrome to *https://rsins.ru* are served a certificate signed by *GlobalSign*, a CA globally recognized by Chrome's root store. Conversely, with RTCA embedded in its trusted root, Yandex seamlessly accepts RTCA certificates without requiring fallback authorities.

This discrepancy suggests that RTCA's inclusion in Yandex's root store facilitates unimpeded connections for Russian users within a government-supported trust framework. At the same time, Chrome's exclusion of RTCA ensures adherence to international CA standards. This browser-dependent CA acceptance thus creates varied levels of certificate validation, underscoring RTCA's potential as a selective tool for user monitoring within Russia's infrastructure.

7. Interpretation of the Results

The findings above help us to consider possible interpretations of the anomaly concerning RTCA, as well as possible implications that this research has for users.

7.1 Interpretation of the Anomaly

The deviation in the certificate chains from the same vantage point suggests two potential scenarios, browser-dependent trust policies, or a possible HTTPS Interception attempt.

7.1.1 Browser-Dependent Trust Policies

The Yandex browser, which aligns closely with Russian government policies, accepts the RTCA certificate by default, thus enabling seamless access to sites secured with RTCA-issued certificates. In contrast, western browsers, like Chrome, enforce a more rigorous certificate validation process, prioritizing certificates from globally recognized CAs like GlobalSign. This divergence in trust policies is particularly significant given the increasing prevalence of state-controlled CAs globally. Yandex's acceptance of the RTCA certificate could indicate a deliberate strategy to facilitate HTTPS Interception, where the browser's trust settings are manipulated to favor domestic CAs over international ones.

7.1.2 Possible HTTPS Interception Attempt?

The substitution of an RTCA-issued certificate in Chrome indicates that specific requests within Russia's internet infrastructure may be routed through intermediaries capable of substituting certificates. If the network infrastructure dynamically assigns RTCA certificates to particular traffic, this could indicate a centralized interception point at the ISP or national gateway level.

An RTCA certificate in the Chrome browser suggests that traffic may be intercepted or proxied at various points within the network. Suppose the network infrastructure can dynamically assign RTCA certificates to specific traffic flows. In that case, it may indicate a centralized interception point at the ISP level or national gateways that manage cross-border traffic. This scenario would

enable state agencies to monitor, log, or manipulate HTTPS traffic under the guise of security and compliance, and aligns with Russia's infrastructure and capabilities, as described in Section 4.

However, at this point, it is not clear how the web server could identify different browsers. Inspecting the network traces captured during the crawl revealed no significant anomalies in the Client-Hello packets. The only difference when comparing the leaf certificates is the longer validity of the one served to Chrome. This difference is not considered a deviation in this study, and could be the result of some form of performance optimization made by the web server. Regional requirements can also be excluded from being categorized as a deviation, since both clients operated in the same region under the same IP address. The only remarkable aspect is the certificate signed by RTCA being served to the Yandex browser only, which is the sole browser to have included RTCA in its trusted root store. This is in contrast to Chrome, which does not trust RTCA, and was probably served a certificate signed by GlobalSign, which is indeed included in Chrome's trusted root store. This fact somewhat excludes load balancing too, which usually involves rotating the request to distribution servers at random, or is dependent on the requester's location.

7.2 Potential Implications for Users

From a state-level perspective, both scenarios described in Section 7.1 require the need for states to protect their citizens in the digital space. However, considering the case of Russia and the state's efforts to build a 'sovereign internet', the discrepancy shown in the findings poses significant implications for users, particularly those under state observation, such as activists.

The selective deployment of RTCA certificates may indicate strategic attempts to intercept encrypted traffic while maintaining an appearance of secure connections. If an HTTPS connection to websites requires using Yandex, they automatically trust RTCA-signed certificates. This may lead to unanticipated data exposure or interception, putting those users' privacy in danger. Furthermore, differences in certificate validation between browsers mean that users may not receive the same level of security assurance.

Both scenarios can be framed by states as efforts to protect citizens in the digital realm. However, given Russia's trajectory of increased network control, the selective use of RTCA certificates in Chrome creates the conditions, and suggests a strategy, for intercepting encrypted traffic while creating an illusion of secure connections. Together, the factors mentioned above underscore a troubling landscape for individuals, particularly those whose online activities the state would have interest in closely monitoring.

8. Conclusion

This report documents our investigation of mass HTTPS Interception attacks carried out during 2023. Due to the timely political context, the resources, and the infrastructure given, we focused our work on Russia's digital landscape. Documented investigations of the country's network makes Russia a prime example of how governments may abuse their power to conduct mass surveillance in the digital space within their borders.

Throughout this work, we assumed Russia's government was a state-level attacker and described a realistic attack scenario under Russia's documented infrastructure within the attacker's reach. We highlighted how this attacker can combine the control of a domestic Certificate Authority (RTCA), the control of a browser (Yandex) and the control of a sub-network (via TSPUs) to potentially carry out mass surveillance and interception of encrypted communications.

8.1 Lessons Learned

Measurement studies on the internet naturally do not provide deterministic results. Deviations within the measurements are to be expected. Configuration failures, network interference, and diverse clients and versions, among others, may lead to temporarily different outcomes within measurement results. However, if deviations, which lead to different security guarantees, are deterministic and can be associated with networks within specific political regimes, their accidental nature must be questioned.

The initial findings during our investigation period did not reveal any specific attacks. However, they demonstrate several anomalies in certificate chains retrieved during our crawls within Russia's and Germany's vantage points. We are still in the process of identifying plausible technical reasons for individual anomalies.

Nevertheless, be it due to misconfigurations in the network or software, or malicious intent, these anomalies result in inconsistent security guarantees, putting people from certain regions, such as those accessing the internet in Russia, at a disadvantage or even in danger of privacy violations. From our perspective, this raises concerns about the integrity and security of HTTPS connections within the regions concerned.

8.2 Presentation of this work

This work and some of its insights have been presented at the following venues:

1. Research paper at USENIX'23 as co-author (FOCI'24 best practical award) (2023).
2. Non-tech talk and panel discussion at Critical Infrastructure Lab (CIL'23) in Amsterdam (2023).
3. Panel discussion at the European Dialogue on Internet Governance (EuroDIG'23) (2023).
4. Research talk at Technical University of Vienna (2023).
5. Press article[71] at Technical University Braunschweig (TUBS) (2023).
6. Talk at SplinterCon'24, Brussels (2024).
7. Research talk at Max-Planck-Institute for Security and Privacy (INET) (2024)

9. Ongoing and Future Work

Despite the time invested, this work is still in its early stages. Analyzing network traffic on a global scale is complex. During our work, we experienced inherent challenges in making accurate distinctions between the origin of anomalies in our data collection. For this, we are already in the process of joining forces with researchers from the domain of Internet Networks and plan to cooperate with widely-used measurement network projects, instead of relying on our hardware. The following describes concrete next steps that we are working on or that are already scheduled.

9.1 Anomaly Analysis

In the process of searching for traces of HTTPS Interception, namely deviations in the certificate chains, it proved challenging to distinguish false positives from an intended and explainable configuration by the web server. Further investigation of the collected network monitoring data is required to understand the circumstances under which such deviations can occur. Various scenarios permit the usage of multiple certificates, which may be technically sound, such as, for example, in the case of load balancing, as described in Section 6.2.1. In contrast, our focus lies on instances of deviation that cannot be attributed to a reasonable technical explanation.

As such, our crawling infrastructure needs to be enhanced with strategies for detecting load balancing to reduce false positives. Although there is never a guarantee, combining different methods can increase the probability of determining the use of load balancing.

Furthermore, at this time, most anomalies described in Section 6.2.1 need to be investigated manually. Therefore, to facilitate more efficient data processing, it is necessary to develop a reliable filtering procedure to exclude false positives.

71 Johns, M. and Dirksen, A. (2023) Information security during war: Consequences of the Ukraine war for internet use. *Technical University Braunschweig*. Available here: https://magazin.tu-braunschweig.de/en/pi-post/information-security-during-war/. Date accessed: 12 December 2024.

9.2 Global Scale

Currently, our infrastructure can only collect and analyze data from, at most, two vantage points. This needs to be expanded to multiple vantage points. Integrating other domestic browsers into the infrastructure could also be interesting, depending on the geolocation of the vantage points. Further, it is essential to note that the characteristics of a network can vary significantly between different countries. These variations are crucial to investigate and consider when analyzing network traffic for traces of specific attacks.

9.3 Detection of HTTPS Interception

In addition to the research presented in this report, we are working on the first proposal towards a detection mechanism that can be used by general internet users, such as a visual indication of a possible HTTPS Interception. Our solution probes the domains requested by the client from another web server outside the attacker's range. This way, the client is served multiple certificates from different sources and can notify the user if they differ. However, this project is still in the early stages of research and development.

Amendment:

Data from a newer experiment shows that the domain *https://rsins.ru* now serves its RTCA-signed certificate to all browsers, including to Chrome.[72] The result is that when a user visits this domain from a browser other than Yandex, they face the warning, as shown by Figure 7. In view of the fact that this domain is used by an important Russian insurance company, we consider this enforcement measure to be critical.

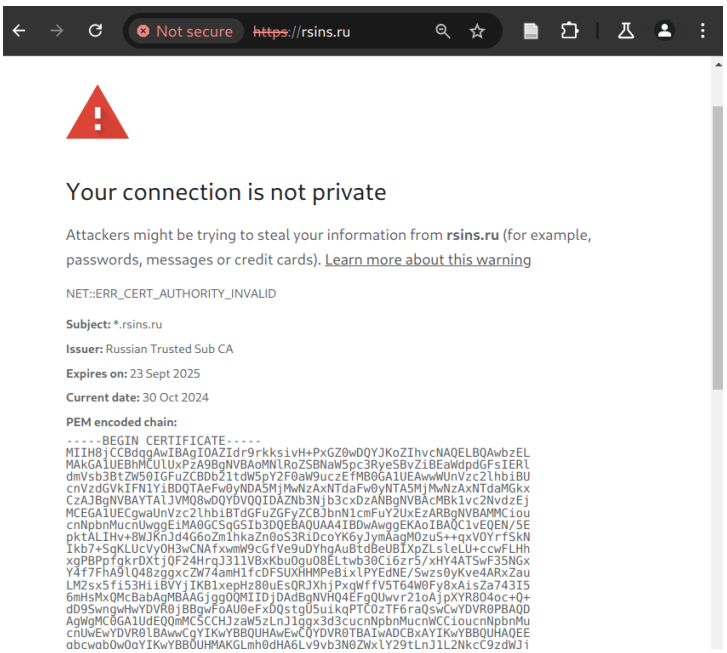

Figure 7: Visiting the domain https://rsins.ru from a Chrome client after Sept. 20, 2024.

72 Date accessed: 30 October 2024.

10. Acknowledgments

This work was done during 2023. It was funded by Open Technology Fund (OTF) under the Information Controls Fellowship Program (ICFP) with Roya Ensafi from Censored Planet acting as my host. I thank Wei Fang and the OTF team for their financial and organizational support during my research. Additionally, I thank Roya Ensafi for her guidance and motivation and for sharing her knowledge.

Front and back cover Designed by upklyak / Freepik

The Threat of State-Level Surveillance Using HTTPS Interception

Alexandra Dirksen
March 2025