Mutual TLS in Practice: A Deep Dive into Certificate Configurations and Privacy Issues

Hongying Dong
University of Virginia
Charlottesville, Virginia, USA
hd7gr@virginia.edu

Yizhe Zhang
University of Virginia
Charlottesville, Virginia, USA
yz6me@virginia.edu

Hyeonmin Lee
University of Virginia
Charlottesville, Virginia, USA
frv9vh@virginia.edu

Kevin Du
University of Virginia
Charlottesville, Virginia, USA
kd5eyn@virginia.edu

Guancheng Tu
University of Virginia
Charlottesville, Virginia, USA
zcn2hb@virginia.edu

Yixin Sun
University of Virginia
Charlottesville, Virginia, USA
ys3kz@virginia.edu

Abstract

Transport Layer Security (TLS) is widely recognized as the essential protocol for securing Internet communications. While numerous studies have focused on investigating server certificates used in TLS connections, our study delves into the less explored territory of mutual TLS (mTLS) where both parties need to provide certificates to each other. By utilizing TLS connection logs collected from a large campus network over 23 months, we identify over 2.2 million unique server certificates and over 3.4 million unique client certificates used in over 1.2 billion mutual TLS connections. By jointly analyzing TLS connection data (e.g., port numbers) and certificate data (e.g., issuers for server/client certificates), we quantify the prevalent use of untrusted certificates and uncover potential security concerns resulting from misconfigured certificates, sharing of certificates between servers and clients, and long-expired certificates. Furthermore, we present the first in-depth study on the wide range of information included in CommonName (CN) and Subject Alternative Name (SAN), drawing comparison between client and server certificates, as well as revealing sensitive information.

CCS Concepts

• **Networks → Network measurement**; • **Security and privacy → Security protocols**.

Keywords

Measurement, Mutual Transport Layer Security, Mutual TLS, Digital certificates, Privacy

ACM Reference Format:
Hongying Dong, Yizhe Zhang, Hyeonmin Lee, Kevin Du, Guancheng Tu, and Yixin Sun. 2024. Mutual TLS in Practice: A Deep Dive into Certificate Configurations and Privacy Issues. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3646547.3688415

IMC '24, November 4–6, 2024, Madrid, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3688415

1 Introduction

Transport Layer Security (TLS) is the de facto standard for encrypting internet communications, with digital certificates functioning as essential cryptographic tools that authenticate the identities of individuals, organizations, or devices, thereby underpinning the security and trustworthiness of network transactions [4].

Many studies [1, 7, 9, 12, 14, 21, 22] have analyzed the TLS certificate ecosystem, focusing primarily on server certificates, which can be collected through active scans of the IPv4 space or Certificate Transparency (CT) logs. However, TLS also supports client authentication through *mutual* TLS, which offers a distinct advantage over traditional TLS by enhancing security through mutual authentication. By facilitating client-side authentication, mutual TLS diminishes the likelihood of unauthorized access or man-in-the-middle attacks, thus mitigating the threat of impersonation attacks. However, the mutual TLS ecosystem has been understudied. While several works [45–48] have conducted preliminary analysis on client certificates, they do not consider server certificates that are also present in mutual TLS.

In this paper, we take a new approach to explore the unexamined aspects of the mutual TLS ecosystem through a *connection-oriented* perspective and integrated analysis of both client and server certificates, by leveraging network traffic data captured from a large campus network.

Research questions. We aim to answer three questions: (1) *How prevalent is mutual TLS and what are the characteristics of mutual TLS connections?* We aim not only to identify common connection characteristics (e.g., ports), but also infer the potential services using mutual TLS, along with patterns of issuers for server and client certificates. (2) *What are the characteristics of certificates used in mutual TLS and are there any security concerns?* Certificates used in mutual TLS connections may exhibit non-standard characteristics, some of which may be concerning, such as the sharing of the same certificate between client and server. We aim to quantify such non-standard behaviors and the scale of their usage. (3) *Do certificates include sensitive information and what are the privacy implications?* While prior works mainly discuss the domain names or IP addresses that are commonly included in `Subject Common Name` (`CN`) and `Subject Alternative Name` (`SAN`), certificates (especially client certificates in mutual TLS) may include non-standard information that could reveal sensitive information about the user or the device, such as personal names or product names. We aim to quantitatively analyze the non-standard information present in CN/SAN fields in client and server certificates.

Dataset. We use SSL logs (with TLS connection data) and X.509 logs (with certificate data) collected from a large campus network from May 2022 to March 2024 (23 months). Our dataset consists of 1.2 billion mutual TLS connections and 5.6 million unique certificates used in mutual TLS. To the best of our knowledge, this is the largest-scale analysis of mutual TLS connections known to date. We perform extensive preprocessing on the dataset to handle complications resulting from real-world data, such as the presence of certificates related to TLS interception. Our preprocessing methodologies are detailed in Section 3.

While sourced from a university setting, our dataset likely reflects mutual TLS patterns applicable to similar environments with rigorous device management and access control. However, this applicability may be limited for residential networks or organizations with different operational frameworks. We discuss the generalizability and limitations of our dataset in details in Section 3.3.

Contributions. We perform an in-depth measurement study using TLS connection data from a campus network spanning over 23 months to understand the *real-world usage* of certificates associated with mutual TLS, as well as investigate the information revealed in the CN and SAN fields. Our key findings are:

1) *Prevalence of mutual TLS:* 38.45% of the certificates presented by servers and 94.34% of those employed by clients[1] are used in mutual TLS connections, with a variety of services.
2) *Concerning practices in certificate usage:* Our study reveals concerning behaviors in mutual TLS, prompting a critical reevaluation of client-side authentication validation procedures in over 13 million connections. These practices include the absence of a valid client issuer in 37.84% of all observed connections, the use of certificates with dummy serial numbers resulting in almost 40,000 collisions within the same issuer, situations where both endpoints employ identical certificates in single connections that involve over 5,000 clients, and the persistent utilization of expired client certificates that have expired for over 1,000 days, with even 42.27% being issued by and utilized for Apple and Microsoft services.
3) *Information revealed in* CN/SAN*:* We uncover a diverse range of information in the CN and SAN fields, many of which include sensitive information. Notably, we identify more than 60,000 client certificates that include personal names or user accounts, posing privacy concerns.

Artifact availability. We unfortunately are unable to provide the original campus network traffic data due to Infosec and IRB rules given their sensitive nature. In the future, our goal is to work with Infosec and IRB to enable the sharing of more intermediate data results in a privacy-preserving way.

2 Background and Related Work

We first describe important concepts that are the most relevant to our study. We then summarize related works and highlight the contribution of our study.

2.1 Background

Mutual TLS and client certificates. TLS supports the capability for clients to authenticate their identity to servers using their own certificates. During the TLS handshake, if a server requests a client certificate after presenting its own, the client responds by transmitting its certificate. This type of mutual authentication is commonly used to provide an additional layer of security, in scenarios such as verifying API endpoints [10] and authenticating among a group of nodes [27]. Consequently, client certificates often contain sensitive private information about the individual that can be observed by the network, posing privacy concerns [25].

Vulnerabilities with insecure certificates. Insecure certificate practices can introduce significant security vulnerabilities. First, the use of expired TLS certificates—whether on the client or server side—raises concerns about certificate validation processes. Without proper validation, connections become vulnerable to man-in-the-middle (MITM) attacks, where attackers can intercept and manipulate communications by presenting their own certificates [20]. Additionally, expired certificates indicate outdated security practices, which can render associated cryptographic algorithms increasingly susceptible to exploitation over time. Second, certificates lacking a valid issuer field compromise the certificate validation process by weakening the security framework dependent on trusted Certificate Authorities (CAs). This deficiency in issuer validation heightens the risk of accepting self-signed or fraudulent certificates, as studied in prior works [5, 36]. Last, using the same certificate at both endpoints poses significant challenges in certificate management, such as difficulties with revocation and renewal. More critically, employing the same private key for both ends of the connection introduces a single point of failure; if this key is compromised, it endangers the security of both endpoints, potentially leading to severe breaches. While such key sharing practice may be permissible in specific enterprise contexts, it is generally discouraged.

Public CAs and Private CAs. Certificates are signed by Certificate Authorities (CAs). If the CA is publicly trusted (i.e., whose root certificate is in major root stores such as Microsoft or Mozilla), then the certificate can be validated through the chain of trust. Conversely, if the CA's certificate is not in the trust store, then the certificate cannot be validated and is commonly referred to as an *untrusted* certificate, or *invalid* certificate in some prior works [9, 14].

In our paper, instead of using "untrusted certificates", we define certificates signed by *public CAs* and *private CAs*, to focus on the issuers of certificates, as follows:

- **Certificates issued by *public CAs***: This encompasses certificates signed by public CAs, whose root (or intermediate) certificates are included in major root stores such as Apple [2], Microsoft [28], and Mozilla Network Security Services (NSS) [30], or in Common CA Database (CCADB) [19].
- **Certificates issued by *private CAs***: This encompasses certificates signed by private CAs, whose root (and intermediate) certificates are not included in the aforementioned root stores or in CCADB. In other words, this category includes certificates (including self-signed) that do not have a valid chain to public CAs.

[1] The remaining 5.66% of client certificates are present in TLS connections with no server certificate, likely related to the university tunneling services.

Subject Common Name (CN) and Subject Alternative Name (SAN). These two fields are commonly used to include information about the certificate owner, which is used in certificate validation. The X.509 standard [4] does not impose strict limitations on the content or format of the CN field. On the other hand, the RFC provides more concrete guidelines of the types of identifiers that can be included in the SAN field, such as domain names, IP addresses, email addresses, and URIs[2]. Due to its ambiguous and untyped format, the use of the CN field has been deprecated for server identification, with a mandate to use the SAN [38, 40]. However, we observe that the SAN, as well as the CN field, often contain a diverse array of information, some of which may raise privacy concerns.

TLS interception. TLS interception occurs when a proxy intercepts and decrypts TLS traffic between a client (e.g., a web browser) and a server (e.g., a website), to inspect for malware or filter content. During this process, the (interception) proxy impersonates the server to the client and establishes a direct connection with the server using its own certificate. As a result, the client does not see the server's actual certificate. Since these proxy certificates do not reflect the true server identity, we exclude them from our analysis, focusing on certificates genuinely used by intended servers, as detailed in Section 3.2.

2.2 Related Work

Our study primarily focuses on mutual TLS connections and the associated certificates, which could be signed by either public or private CAs. We describe the limited number of studies that have analyzed client certificates or certificates issued by private CAs, and highlight how our approach diverges from these works.

Client certificates. Xia *et al.* and Yin *et al.* conducted several short studies [45–48] on TLS client certificates. They examined certain characteristics of client certificates such as validity periods and issuer organizations, along with brief information on TLS connections such as TLS versions and port numbers. They also mentioned the presence of sensitive information in CN and SAN fields in client certificates, such as personal names or device types, but did not perform any quantitative evaluation.

Several works [16, 44] have utilized client certificates to track users by leveraging details within the certificates (e.g., serial number) along with related connection information (e.g., IP address and port). These studies primarily used client certificates as a tool for tracking, rather than focusing on the characteristics of the certificates themselves.

Our work diverges from these works by (1) focusing on the mutual TLS connections and examining both server and client certificates together, (2) identifying concerning usage of certificates in mutual TLS that have not been previously discussed, such as the sharing of certificate between server and client, and (3) performing an in-depth and quantitative analysis on CN and SAN fields.

Certificates issued by Private CAs. While there exist many studies on valid TLS certificates signed by publicly trusted CAs, only a few studies [9, 14] analyzed untrusted certificates signed by private CAs. Chung *et al.* [9] conducted active scanning of the IPv4 space on port 443 and collected server certificates. Farhan *et al.* [14] expanded the dataset to include Certificate Transparency (CT) logs. They highlighted unique characteristics of invalid server certificates, such as validity periods and issuer diversity.

These two studies only examined server certificates and did not have any visibility into the TLS connections utilizing such certificates. On the contrary, our study focuses on mutual TLS connections and examines both client and server certificates jointly.

3 Dataset

We first describe our data collection and then delve into the details of data processing and enrichment.

3.1 Data Collection

We cooperate with university's information security department and conduct passive data collection on university's border gateway from May 1st 2022 to March 31st 2024 (a total of 23 months). The raw border traffic is mirrored to a cluster in the secure DMZ (demilitarized zone) and subsequently processed using Zeek software [49], a widely adopted security monitoring tool, generating a collection of `SSL.log` and `X509.log` files:

- Zeek `SSL.log` comprehensively captures network traffic employing Transport Layer Security (TLS) protocols by utilizing dynamic protocol detection (DPD) [50] techniques, rather than solely relying on standard ports. Consequently, our collected `SSL.log` differs from prior works that solely depend on active scanning of HTTPS traffic on port 443 [9, 14], encompassing a wide variety of network traffic utilizing TLS protocol. Additionally, `SSL.log` provides detailed information of TLS connections, including the IP, port, the server name (SNI) of the connection, the certificate chain information, and the success of connection establishment.
- Zeek `X509.log` complements `SSL.log` by extracting and parsing the intricate details of certificates exchanged during TLS negotiations, such as certificate serial number, issuer, subject, validity time, and encryption parameters. Each certificate in `X509.log` is linked to `SSL.log` through unique IDs recorded during the authentication process.

Combining the two types of log, our study focuses not only on the characteristics of certificates, but also the usage of certificates in TLS connections.

Ethic considerations. Our data collection and usage have undergone approval from University Infosec and Institutional Review Board (IRB). We discuss ethic considerations in detail in Appendix A.

3.2 Data Enrichment and Statistics

3.2.1 Methodology. We describe methodologies to identify and extract certain characteristics from the certificates. Note that our analysis is conducted using established TLS connections.

Mutual TLS. We identify a certificate participating in mutual TLS when its recorded TLS connection includes both the server certificate chain and the client certificate chain.

Server and client. We determine whether a certificate is intended for server authentication or client authentication by examining the

[2] The CA/Browser (CA/B) forum imposes stricter guidelines [6], requiring that the SAN field has to contain at least one domain name or IP address.

Certificates	Total	Mutual TLS	
	Num.	Num.	%
Total	9,472,584	5,629,861	59.43
Server	5,915,995	2,274,748	38.45
- Public CA	3,176,415	6,942	0.22
- Private CA	2,739,580	2,267,806	82.78
Client	3,556,589	3,355,113	94.34
- Public CA	26,011	22,677	87.18
- Private CA	3,530,578	3,332,436	94.38

Table 1: Number of unique certificates. The ratio is calculated against the total number of certificates in each category.

certificate chain logged in Zeek `SSL.log`. If the certificate appears in the server-side certificate chain, we label it as a server certificate, and conversely for client certificates. Notably, we observe instances where certificates are utilized for both authentication purposes. Further elaboration on this phenomenon is provided in Section 5.2.

Public and private. Zeek software utilizes Mozilla Network Security Services (NSS) [30] for certificate chain validation. We enhance this validation process by incorporating other prominent trust stores such as Apple [2], Microsoft [28], and Common CA Database (CCADB) [19]. A certificates is deemed to be issued by public CAs when its root or intermediate certificate, or its issuer, is listed in at least one of the aforementioned major trust stores [2, 19, 28, 30]. Certificates not meeting the above criteria is considered as issued by private CAs.

Internal and external. We employ university's IP subnets to distinguish whether the certificate is used within or outside of the university's network. The data is then utilized to determine the direction of TLS connections (i.e., inbound or outbound) in our subsequent analysis.

Interception certificates. Software or applications installed on personal devices may be configured to perform TLS interception for security purposes when accessing the campus network. In instances of TLS interception, the encrypted TLS traffic undergoes decryption and subsequent re-encryption by an intermediary using a private key. This process alters the issuer of the server-side certificate, thereby significantly biasing our analysis. Therefore, we need to exclude certificates of TLS interception from our analysis. To achieve this, we first filter connections wherein the issuer of the server-side leaf certificate is *not* found in major trust stores [2, 19, 28, 30]. We then utilize Certificate Transparency (CT) logs [11] to find the original issuer of corresponding domain and compare it with the logged certificate. Finally, we conduct manual investigation of cases where the logged certificate issuer is different from the issuer in CT logs. Consequently, we identified 186 issuers indicative of TLS interception and subsequently excluded a total of 871,993 (8.4%) certificates from our dataset.

3.2.2 Certificate statistics. Our analysis focuses on the leaf certificates in our datasets. Table 1 presents the statistics for unique leaf certificates after filtering out interception cases. In total, 9.5 million unique leaf certificates are collected, of which 59.43% are engaged in mutual TLS authentication. Although only 0.22% public CA-issued server certificate are observed during mutual TLS authentication, a significant 82.78% of server certificate issued by private CAs are deployed for mutual authentication. Moreover, nearly 94.34% collected client certificates are used in mutual TLS authentication, whereas the remaining 5.66% of client certificates are present without any server certificate in the connection, likely attributed to the university tunneling services. These numbers suggest a considerable volume of certificates used in mutual TLS connections.

3.3 Dataset Generalization and Limitation

While our dataset is derived from a university setting located in the United States, the observed patterns and types of mutual TLS applications are likely generalizable to other similar contexts, as specified below:

1) The campus network, which supports around 10,000 faculty and staff and over 23,000 students, handles a diverse array of devices and services, with over 30% of inbound mutual TLS traffic related to device authentication and access control systems.
2) The campus network includes significant traffic volume involving a public medical center, which accounts for 64.9% of the inbound mutual TLS traffic.
3) Outbound traffic mainly involves cloud services, security services, and email protocols, with over 6% of outbound mutual TLS connections related to email and more than 68% of external servers associated with popular cloud and security providers such as Amazon Web Services (AWS) and Rapid7.

These factors suggest that mutual TLS patterns in our dataset are likely applicable to other similar environments, including hospitals, educational institutions, and enterprises, which similarly prioritize strong device management and access control. However, this generalizability may be limited for residential networks or entities with different operational frameworks, which might exhibit mutual TLS behaviors not captured in our study. Additionally, the inability to capture certificates in TLS 1.3 connections—constituting 40.86% of all TLS connections, involving 25.35% and 32.23% of all server and client IP addresses—due to default encryption means that we cannot ascertain whether these connections involve mutual TLS. Thus, our analysis may not fully reflect the entire spectrum of mutual TLS certificate usage.

4 Mutual TLS

The primary benefit of mutual TLS compared to traditional TLS lies in enhanced security through mutual authentication. By enabling client-side authentication, mutual TLS reduces the risk of unauthorized access or man-in-the-middle attacks and mitigates the risk of impersonation attacks. We now take a closer look into connections with mutual TLS.

Inbound and outbound. Our network traffic is captured at the perimeter of the university network, encompassing bidirectional traffic flows entering and exiting the institution. Subsequently, the data is segregated into two primary categories: inbound and outbound traffic, in subsequent analysis.

Connections as a metric. We use the number of TLS connections as our primary metric due to limitations in accurately representing client counts through IP addresses, as clients in the campus network are extensively using Network Address Translation (NAT). Additionally, a single client may utilize multiple certificates across various TLS connections, even to the same server, complicating

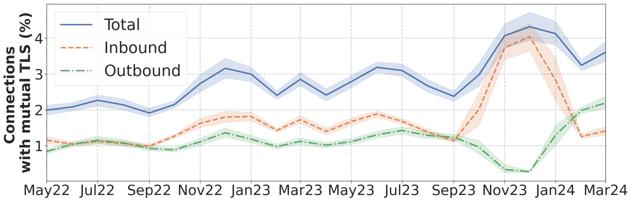


Figure 1: Percentage of TLS connections that employs mutual TLS authentication.

analysis based on client percentages. To mitigate potential distortions from a few clients disproportionately affecting the results, we present the number of client IPs as an estimate of distinct clients in our analysis.

4.1 Prevalence of Mutual TLS

We first take a look at the adoption of mutual TLS and identify possible services utilizing mutual TLS authentication to secure their connections.

How many connections use mutual TLS? We start our analysis by examining TLS connections implementing mutual TLS authentication between May 2022 to March 2024. Over this timeframe, we observe a near doubling in the overall adoption of mutual TLS authentication, rising from 1.99% to 3.61% of total TLS connections (see Figure 1). Specifically, on a daily basis, we observe 1.26 million mutual TLS connections at the beginning of our study (May 2022) and 2.36 million connections at the end of our data collection period (March 2024). Particularly, for inbound traffic, we note a nearly twofold increase in traffic to the university health services from October 2023 to December 2023, which contributed to the observed surge depicted in Figure 1. In contrast, outbound traffic to domains such as Rapid7 disappeared during the same period, likely due to changes in network topology, resulting in the decline from October 2023 to December 2023, as shown in Figure 1.

What services are carried by mutual TLS? We next examine the prominent services with mutual TLS authentication. We identify these services by searching port registry information in Internet Assigned Numbers Authority (IANA) [17] and by examining the logged TLS connection and corresponding certificate information.

Table 2 shows the top 5 services for both inbound and outbound traffic. Note that *Corp.-Miscellaneous* suggests that several companies, including Amazon FireHose and Mixpanel, use port 3128 for various services.

While HTTPS is the primary protocol for both mutual and non-mutual TLS connections, mutual TLS connections show a lower ratio of HTTPS compared to non-mutual TLS connections. Additionally, inbound traffic shows a lower ratio of HTTPS for both mutual and non-mutual TLS. Outbound traffic shows a similar distribution across mutual and non-mutual TLS, where HTTPS is the majority, amongst other protocols such as Message Queuing Telemetry Transport (MQTT) [31] or Simple Mail Transfer Protocol (SMTP) [23]. However, inbound traffic shows quite different characteristics for mutual TLS. We notice that some specific services utilize dedicated ports for their service, particularly in inbound traffic with mutual TLS. For instance, 24.89% of inbound traffic with mutual authentication is associated with the 'FileWave' [15] device management service on port 20017, and ports in the range 50000 to 51000 are used for the 'Globus' [32] data transfer service. In contrast, mutual TLS traffic shows fewer such cases due to the dominance of HTTPS (85% for inbound and 99% for outbound). Additionally, we observe the Lightweight Directory Access Protocol (LDAP) [41] (6.36%), which is primarily utilized for university access control.

(*Takeaway*) We observe an increasing usage of mutual TLS authentication over the 2-year period, and identify varying services besides the most popular HTTPS.

4.2 Mutual TLS Connection

We now delve into the specifics of certificates and servers in mutual TLS connections. We begin by introducing our methodology for categorizing traffic and certificate issuers.

Methodology. We extract the Top-Level-Domain (TLD) and Second-Level-Domain (SLD) from the `Server Name Indication` (SNI) field for each TLS connection. In instances where the SNI is absent, we resolve server information by investigating the `SAN DNS` and `Subject CN` fields of both the server-side and client-side leaf certificates. We further classify issuers of client-side certificates into the following categories by examining the presence of either the issuer of the leaf certificate (referred to as intermediate certificate) or the issuer organization in CCADB or major trust stores. In cases where issuers are classified as *Private*, we conduct fuzzy matching and necessary manual validation on the issuer organization string to enhance the precision of issuer groupings. Instances that remain unverified are designated into the *Other* category. Additionally, *MissingIssuer* signifies the absence of a value for the issuer organization:

- *Public*: issuer or issuer organization of the certificate in either CCADB or major trust stores.
- *Private - Corporation*: issuer organizations recognized as corporation names.
- *Private - Education*: issuer organizations recognized as universities and schools.
- *Private - Government*: issuer organizations recognized as government.
- *Private - WebHosting*: issuer organizations recognized as companies providing web hosting services.
- *Private - Dummy*: issuer organizations recognized as software or protocol default strings (further discussed in Section 5.1.1).
- *Private - Others*: issuer organizations not recognized during the fuzzy matching.
- *Private - MissingIssuer*: missing issuer organizations.

Inbound traffic. Inbound traffic exhibits less variability on the server side due to its predominantly university-hosted destinations. In light of this observation, we classify servers based on the extracted SLDs into the following categories:

- *University Health*: comprising domains associated with the university health system.
- *University Server*: including domains of general purpose operated by the university.
- *University VPN*: encompassing domains served for the university's VPN system.
- *Local Organization*: covering servers managed by local organizations collaborating with the university.
- *Third Party Service*: servers provided by external services.
- *Globus*: Globus file transfer servers.

	Server ports with mutual TLS authentication						Server ports without mutual TLS authentication					
	Inbound			Outbound			Inbound			Outbound		
Rank	Port	%	Service	Port	%	Service	Port	%	Service	Port	%	Service
1	443	63.60	HTTPS	443	83.17	HTTPS	443	85.18	HTTPS	443	99.15	HTTPS
2	20017	24.89	Corp.- FileWave[1]	8883	3.69	MQTT over TLS	25	2.35	SMTP	993	0.44	IMAPS
3	636	6.36	LDAPS	25	3.38	SMTP	33854	2.26	Corp.- DvTel[5]	8883	0.05	MQTT over TLS
4	50000-51000	1.17	Corp.- Globus[2]	465	3.32	SMTPS	8443	2.22	HTTPS	25	0.04	SMTP
5	9093	0.26	Corp.- Outset Medical[3]	9997	1.48	Corp.- Splunk[4]	52730	1.98	Univ.- Unknown	3128	0.03	Corp.- Miscellaneous

[1] FileWave provides multi-platform endpoint management solutions.
[2] Globus provides data transferring and management services.
[3] Outset Medical manufactures dialysis devices.
[4] Splunk provides cloud and data services.
[5] DvTel provides video surveillance solutions.

Table 2: Prominent services with or without mutual TLS authentication, with HTTPS being the most prominent.

Server association	%. connections	%. clients	Client certificate issuer			
			Primary	%. clients	Secondary	%. clients
University Health	64.91	41.10	Private - Education	99.96	Public	0.94
University Server	30.55	5.00	Private - MissingIssuer	95.84	Public	3.70
University VPN	0.30	14.73	Private - Education	99.99	Public	0.01
Local Organization	2.53	2.20	Public	96.62	Private - Corporation	1.32
Third Party Services	0.31	0.39	Private - Others	47.95	Public	37.25
Globus	0.06	<0.01	Private - Education	93.83	Private - Others	6.17
Unknown	1.34	36.58	Private - MissingIssuer	87.34	Private - Others	12.39

Table 3: Proportion of connections, clients, and issuers of client certificates by server association in inbound traffic.

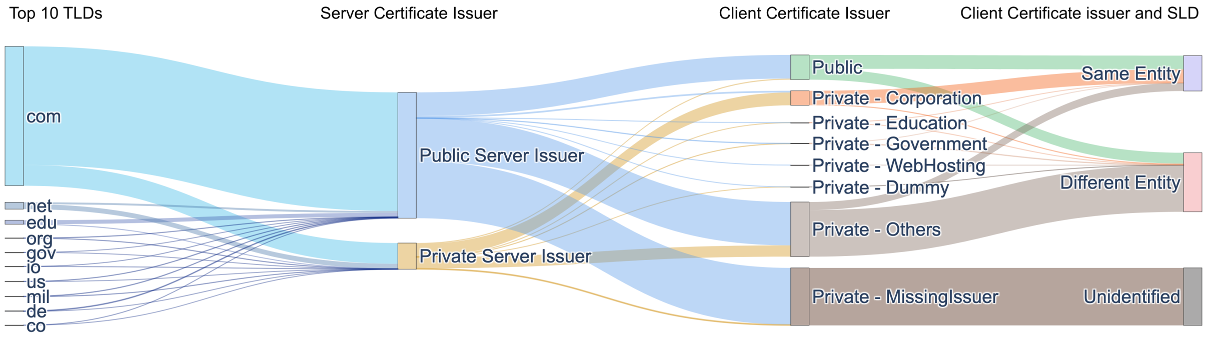


Figure 2: Issuers of server-side and client-side certificates for outbound traffic with the most prevalent TLDs. Each flow represents a mutual TLS connection with a valid SNI field in the ClientHello.

- *Unknown*: domains lacking valid server information.

Outbound traffic. In contrast, the *outbound* traffic exhibits considerable diversity from the server perspective. Consequently, rather than categorizing SLDs, we categorize outbound traffic according to the TLDs of the servers. Furthermore, we refine the classification of server-side leaf certificates into *Public* and *Private* categories (see Section 3.2). Additionally, we employ fuzzy matching on client-side certificate issuers and SLDs to ascertain whether the domain owner and the certificate issuer belong to the same entity.

4.2.1 *Mutual TLS in inbound traffic.*

We show the statistics of client certificates employed in inbound connections in Table 3. Most connections employing client authentication correspond to university-based servers, with client certificates issued by the respective universities. This is comprehensible given that university services typically restrict access to authorized clients. Although only a minority of connections with missing SNIs necessitate mutual trust establishment, these connections encompass more than one-third of clients, with the majority of client certificates in these instances lacking a valid issuer field. Such deficient issuing practices render the certificate issuer unidentifiable, potentially exposing the connections to man-in-the-middle (MITM) attacks.

4.2.2 *Mutual TLS in outbound traffic.*

We show the issuer categories of both server-side and client-side certificates for outbound mutual TLS traffic with the most prevalent TLDs in Figure 2. A significant portion of the connections involve server certificates issued by public-trust entities and client certificates issued by private entities, albeit belonging to a different domain entity than the server. While there is a diverse distribution of server associations among these connections, a majority of the SLDs are associated with cloud service providers, with `amazonaws.com` (28.51%), `rapid7.com` (27.44%), and `gpcloudservice.com` (13.33%) being the most prevalent. The client certificate issuers in these connections predominantly represent individual private entities, resulting in a discrepancy between the issuer entity and the corresponding SLD.

	Client certificate				Server certificate			
	Dummy issuer organization	**#. involved servers**	**#. involved clients**	**#. conn**	**Dummy issuer organization**	**#. involved servers**	**#. involved clients**	**#. conn**
In.	Default Company Ltd	Local Organization	22	31	-	-	-	-
	Internet Widgits Pty Ltd		21	95				
Out.	Unspecified	com, edu, org, gov, net	452	566,996	Internet Widgits Pty Ltd	com, edu, io	511	3,689
	Internet Widgits Pty Ltd	com, io	73	69,069	Default Company Ltd	com, edu, cn, co	147	331
	Default Company Ltd	cn, top	2	17	Acme Co	com	20	26

Table 4: Certificates with dummy issuer in mutual TLS connections. Note that for inbound traffic (denoted as In.), servers are categorized using SLDs, but for outbound traffic (denoted as Out.), servers are refined based on TLDs.

In addition, we note that 45.71% of connections, in cases where server-side certificates are issued by public-trust CAs, utilize client certificates issued by private entities that lack a valid issuer field. This pattern again prompts a critical reassessment of the adequacy of client-side authentication validation procedures.

(***Takeaway***) The majority of client certificates are issued by a variety of private issuers, with a significant number associated with individual private entities utilizing cloud services in outbound traffic. Notably, 37.84% of all outbound client certificates lack a valid certificate issuer, raising concerns regarding the adequacy of the client authentication validation process.

5 Mutual TLS Certificate Practices

The widespread adoption of public audit mechanisms, such as Certificate Transparency (CT), play a pivotal role in swiftly identifying and rectifying misconfigured certificates within server deployments. However, when considering client authentication, the regulatory landscape is notably sparse. This dearth of oversight may inadvertently pave the way for misconfigurations or misuse, consequently amplifying the potential for exploitable vulnerabilities within the system. In this section, we delve into certificates presented in mutual TLS connections, with an emphasis on the client authentication.

Duration of activity. To monitor the usage patterns of certificates or connections, we define ***duration of activity*** as the interval between the initial observation date and the latest observation date for each certificate or certificate-involved connection. For example, if a client certificate is observed once per month, its activity duration over a one-year observation period would encompass the entire year.

Connection tuple. To provide further insight into connection-wise statistics, we define a ***connection tuple*** as a unique combination of (client, client certificate, server, and server certificate) in mutual TLS connections.

5.1 Dummy Information in Certificates

While tools and protocols such as OpenSSL and Let's Encrypt ACME (Automated Certificate Management Environment) streamline the certificate issuance process [26, 33], there persists uncertainty regarding users' proficiency in accurately configuring digital certificates upon request.

5.1.1 Dummy Issuers.

Within our dataset, we observe the presence of certificates retaining the default dummy organization name of the respective software or protocols used, as demonstrated in Table 4 for client-side and server-side certificates. In particular, all of these TLS connections with default-configured certificates are successfully established.

Specifically, we observe certificates associated with dummy issuers used by both endpoints in the same outbound connections. These connections involve three server associations, all linked to the issuer 'Internet Widgits Pty Ltd' for both client and server certificates, affecting 17 clients with the highest duration of activity surpassing 600 days. We show details in Appendix B.

In addition to the risk of inadequate validation stemming from dummy issuers, another significant concern associated with dummy certificates is their lack of unique cryptographic parameters tailored to the server. Instead, they utilize outdated algorithms or generic keys that are widely recognized, rendering them vulnerable to potential security breaches such as man-in-the-middle attacks or eavesdropping. Alarmingly, among all client certificates with dummy issuer organizations, we identify 3 issued by the OpenSSL dummy issuer 'Internet Widgits Pty Ltd' that implement certificate version 1.0, involving 154 unique connection tuples; 13 issued by the dummy issuer 'Unspecified' that utilize a 1024-bit RSA key, involving 83 unique connection tuples. Note that the National Institute of Standards and Technology (NIST) has disallowed the use of 1024-bit keys after 31 December 2013 [13].

5.1.2 Dummy Certificate Serial Numbers.

The TLS certificate serial number is an identifier assigned by the issuer to each certificate. According to RFC 5280, the serial number must be unique for each certificate issued by a specific CA [4]. Incorporating randomness in serial numbers prevents predictability and potential vulnerabilities [42]. Our analysis unveils that numerous certificates employed in mutual TLS are with the identical dummy serial number within the same issuer's scope. This observation applies to both client and server certificates, detailed as follows.

Inbound traffic. A total of 1,126 clients are involved in inbound connections where at least one endpoint utilizes a certificate with serial number collisions. The most prevailing serial number with collisions found in inbound certificates is `00`, utilized by 6 distinct private issuers. Notably, the highest occurrence is observed with 38,965 unique client certificates and 38,928 unique server certificates issued by the same entity, 'Globus Online'. All certificates with the serial number `00` from 'Globus Online' are observed from both the client and the server within the same connection, totaling 7.49 million connections and 798 clients across our whole study period (we further identify that the same certificate are shared by both endpoints in the same connection in Section 5.2.1). The noticeable difference between the count of unique certificates and the number of clients arises from the certificate re-issuance, as the validity period of these certificates spans only 14 days. More specifically, all these connections use the SNI 'FXP DCAU Cert', and employ certificates issued by 'Globus Online', with the issuer CN listed as 'FXP DCAU Cert'.

Additionally, other frequently encountered serial numbers such as `01`, `02`, and `03` also experience collisions within the same issuer, all of which are associated with servers categorized as *Local Organization*. A notable finding pertains to the consistent issuance of all certificates with the same serial number `024680` by a private issuer named 'ViptelaClient', regardless of whether they are designated for client-side or server-side utilization. Furthermore, it is worth mentioning that certificates with such serial numbers typically exhibit short validity periods, with the majority lasting less than 15 days.

Outbound traffic. For outbound traffic, 14,541 clients are associated with connections where at least one endpoint uses a certificate with serial number collisions. Among these, 4,593 (31.59%) clients are involved in 2.76 million connections where both endpoints utilize certificates with serial number collisions. The prevalent serial numbers exhibiting collisions in outbound traffic mirror a similar pattern observed in inbound traffic: `00` and `01`. In addition to instances where outbound connections utilize client and server certificates both issued by 'Globus Online' with the identical serial number `00` (including 8,260 unique certificates employed in 9,600 connection tuples), we identify another private issuer, 'GuardiCore', which implements mutual TLS and issues certificates for both clients and servers. All client-side certificates from 'GuardiCore' share the identical serial number `01`, while all server certificates have the same serial number `03E8`. These corresponding 904 connections, associated with servers lacking SNIs, persist throughout our study period, involving 57 client certificates and 43 server certificates in 418 unique connection tuples. Unlike 'Globus Online', the certificates issued by 'GuardiCore', irrespective of their designation as client-side or server-side, have a longer validity period that exceeds 2 years.

5.2 Certificate Sharing

Although it is technically feasible for a TLS certificate to serve as both a server and a client certificate, such practice is not advisable due to the heightened risk of compromising the private key, as it becomes accessible on both ends.

5.2.1 *Sharing in the Same Connection.*

We observe the occurrence of both endpoints presenting the same certificate in one connection, as detailed in Table 5. It is noteworthy that two distinct usage patterns emerge: some entities choose to privately issue a single certificate for both server and client purposes, while others, utilizing trusted CAs for server certificates, also employ these certificates for client authentication. In addition, such practice involves 7.49 million and 5.93 million connections, for inbound and outbound, respectively.

5.2.2 *Sharing in Different Connections.*

We identify 1,611 certificates shared across servers and clients in distinct connections. By calculating the occurrence of each specific certificate used as either a client or server certificate within /24 subnets, we observe that while 99% of certificates presented by clients are shared across 7 subnets, the number of subnets increases to 43 for 99% of certificates used by servers, as detailed in Table 6. We note that the most prevalent issuers of these certificates consist of intermediate certificates responsible for issuing server-side certificates from Let's Encrypt (51.58%), DigiCert (14.34%), and Sectigo (7.95%). This indicates a similar usage pattern discussed before: utilizing trusted server certificates for client authentication.

	SLD	Certificate issuer	#. involved clients	Duration of activity (days)
In.	- (missing SNI)	Globus Online	699	700
Out.	tablodash.com	Outset Medical	4,403	700
	- (missing SNI)	Globus Online	105	699
	psych.org	American Psychiatric Association	2	424
	splunkcloud.com	Splunk	4	114
	leidos.com	IdenTrust[1]	52	554
	acr.og	GoDaddy.com, Inc[2]	24	364
	sapns2.com	GoDaddy.com, Inc[2]	1	5
	bluetriton.com	DigiCert Inc[3]	1	1
	gpo.gov	DigiCert Inc [4]	1	1

[1] TrustID Server CA O1.
[2] GoDaddy Secure Certificate Authority - G2.
[3] GeoTrust TLS RSA CA G1.
[4] DigiCert SHA2 Extended Validation Server CA.

Table 5: Details on certificates shared by the client and the server in the same connection. The gray-colored area denotes those with publicly trust issuers.

# of subnets	Quantile			
associated w/	50th	75th	99th	100th
Server	1	1	7	217
Client	1	2	43	1,851

Table 6: The number of \24 subnets with certificate presences in the server and client authentication, respectively.

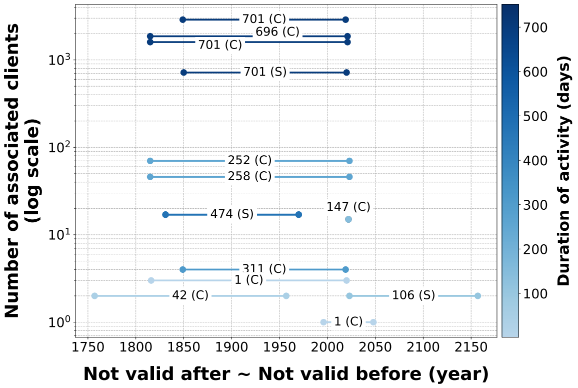
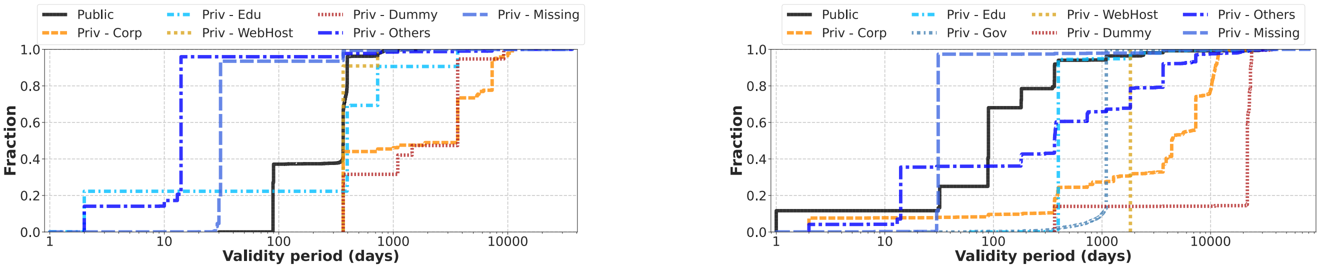


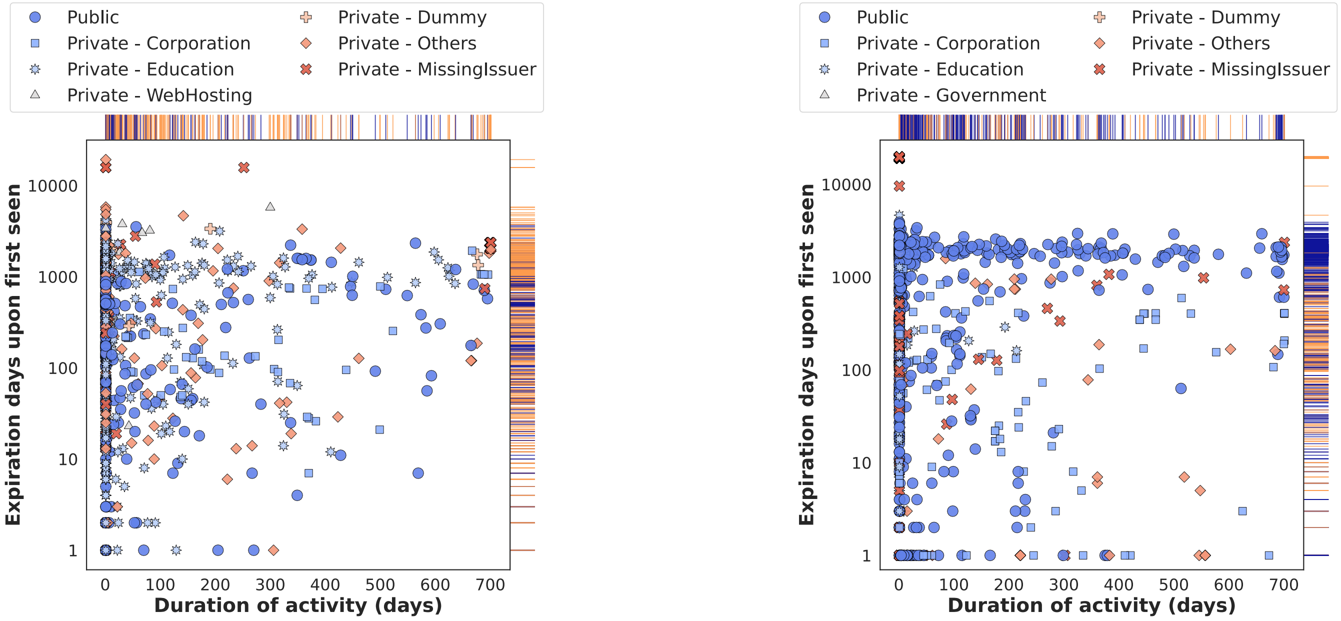
Figure 3: The `not_valid_before` and `not_valid_after` dates, the number of associated clients, and the duration of activity for each certificate. Each line represents the period from `not_valid_after` to `not_valid_before`; note that all observed certificates have `not_valid_after` that precede their `not_valid_before`, except one which has the same timestamp for both with the year 2022. The duration of activity and whether the certificate is a server ('S') or a client certificate ('C') are shown above each line.



(a) Inbound client certificates.

(b) Outbound client certificates.

Figure 4: Issuers and corresponding validity periods (in days) of client certificates.



(a) Inbound client certificates.

(b) Outbound client certificates.

Figure 5: Duration of activity of expired client certificates. Marginals show the distribution of *Public* and *Private* client certificate issuers, where color blue denote *Public* and color orange denote *Private*.

5.3 Certificate Validity Period

The TLS certificate validity period denotes the timeframe in which the certificate remains valid, defined by specific dates and times indicated in the `not_valid_before` and `not_valid_after` fields of the certificate. Public CAs typically adhere to industry standards and guidelines, which stipulate validity periods usually spanning from a few months to several years. In contrast, private CAs, for both server- and client-side, often establish their own internal policies governing certificate validity periods. These periods are oftentimes longer compared to those of public CAs [9].

5.3.1 Certificates with Incorrect Dates. Besides the aforementioned dummy certificates, we also observe certificates with incorrect dates used in mutual TLS, as shown in Figure 3, with details presented in Appendix C. All of these misconfigured certificates have a timestamp specified in `not_valid_before` that does not precede the timestamp in `not_valid_after`, and they are all seen within successfully established connections.

In particular, we identify connections with two server associations, one under SLD `idrive.com` and the other with a missing SNI, that utilize certificates with incorrect dates at *both* endpoints. Each pair of these certificates (client and server) shares the same private issuer, involving 718 clients for over 700 days and 17 clients for over 470 days, respectively. Details are provided in Appendix C.

5.3.2 Validity Periods. Excluding those with incorrect dates, we show the issuers and validity periods of client-side certificates for both inbound and outbound traffic in Figure 4. We observe that certain client certificates captured in outbound traffic exhibit significantly longer validity periods compared to those observed within inbound traffic. In particular, there are 7,911 client certificates exhibiting validity periods ranging between 10,000 days (about 27 years) and 40,000 days (approximately 110 years). Among these, 50 certificates are issued by public authorities, while the remaining 7,861 are issued by private entities, predominantly involving those with empty issuer field (45.73%), corporations (37.58%), dummy organizations (7.61%). The most prevalent TLD associations of these client certificates include com (32,84%), net (35.38%), and those with empty SNI fields (28.06%). Moreover, we identify 1 client certificates with a remarkably extended validity period of 83,432 days (around 228 years). This certificate, issued by a private corporation, is associated with servers under the SLD `tmdxdev.com`.

Such practices in the issuance of client certificates may significantly extend the exposure window of private keys and impede security enhancements, rendering TLS connections susceptible to exploitation by malicious actors.

5.3.3 Expired Certificates. As certificates reach the termination of their validity period, they become expired, rendering them unsuitable for facilitating secure connections between clients and servers. However, we have detected instances of expired client certificates being presented in successfully established TLS connections, occurring within both inbound and outbound traffic streams.

Upon our initial identification of these expired certificates, we proceeded to monitor their activity. Leveraging our previously defined *duration of activity*, we show the actual usage of these expired client certificates in Figure 5.

Among all inbound connections with expired client certificates, the most common server associations are *University VPN* (45.83%), *Local Organization* (32.79%), and *Third Party Service* (15.38%). Unlike the broadly distributed certificates shown in Figure 5a, Figure 5b reveals a cluster of certificates for outbound traffic issued by public-trust CAs with similar expiration periods of around 1,000 days upon our initial observation. Within this cluster, 337 out of 339 certificates are issued by Apple and are associated with servers under the SLD `apple.com`. The remaining two certificates are issued by Microsoft and linked to servers under `azure.com` and `azure-automation.net`, respectively.

(*Takeaway*) While mutual TLS is intended to bolster the security of TLS connections, our research uncovers troubling client authentication practices, affecting 13 million connections. Additionally, certificates with dummy issuers or serial numbers compromise security by increasing the risk of accepting fraudulent certificates. Moreover, using the same certificate for both endpoints—while sometimes permissible in certain enterprise settings—generally undermines security and increases the risk of private key compromise.

6 What do certificates reveal?

Now, we examine the *types of information* included in the certificates used by mutual TLS connections, focusing on the `Subject CN` and SAN fields. Several specifications [4, 38, 40] provide guidelines (specifically for SAN) on the permissible types of information, including domain names, IP addresses, email addresses, and URIs. Our analysis assesses whether certificates adhere to these guidelines by including standard types of information or if they contain other types of non-standard information. Additionally, any information in certificates can be observed by the network (unless using TLS 1.3), which could pose privacy concerns if sensitive information about the certificate owner is included in the CN and SAN fields.

6.1 Methodology

We describe our methodology to process information in CN and SAN fields, and the scope of our analysis.

6.1.1 Classifying Types of Information. We first classify information types with specific or well-known formats based on the specifications [4, 38, 40] or unique formats associated with the university where the data is collected. We create regular expressions accordingly to identify the following cases:

- **Domain name**: We use the Python *tldextract* package [24], which leverages the Public Suffix List [29].
- **IP address**: We determine IP addresses using regex matching (e.g., 1.2.3.4) and the Python *ipaddress* package [18].
- **MAC address**: Medium Access Control (MAC) addresses are identified using regex matching based on their standard formats (e.g., 12:34:56:AB:CD:EF).
- **SIP address**: Some certificates include Session Initiation Protocol (SIP) addresses[3] in their CN or SAN fields. We identify them using regex matching (e.g., sip:*sip-address*).
- **Email address**: Email addresses are detected based on regex patterns (e.g., including the '@' character).
- **User account**: This type is specific to user IDs assigned to each member of the university where our data is collected. Such user IDs have a predefined format consisting of alphanumeric characters. We use regex matching to identify such IDs, and further check whether issuer fields contain names of CAs managed by the university.
- **Localhost**: Certificates that include 'localhost' or 'localdomain' in their CN or SAN fall under this type.

However, aside from the commonly seen types, a significantly number of certificates essentially have free text information in CN and SAN, making it impossible to classify using regular expressions. To tackle this challenge, we employ the spaCy's *en_core_web_trf* pre-trained model [43], a transformer pipeline that has been trained using datasets including OneNotes and WordNet. The Named Entity Recognition (NER) function of this model categorizes text into several labels, including 'PERSON', 'ORG', 'PRODUCT', and 'DATE'. We then perform additional manual verification:

- **Personal name**: We initially identify potential personal names if CN or SAN is labeled as 'PERSON' by spaCy's pre-trained model. This is followed by a manual review process to filter out any misclassified entries that are not actual personal names.
- **Product**: Similarly, we identify potential product names if they are labeled as 'PRODUCT' by the pre-trained model, and then go through a manual review process.
- **Organization**: First, we classify entries as 'ORG' by the pre-trained model. Additionally, we leverage a publicly available list of company names [3, 37] to further classify unrecognized entries from the model. We generate word vectors for these company names and compare them to the entries (in CN or SAN) using cosine similarity. Entries with a similarity score exceeding 0.9 are classified as containing a company name. Finally, the classification results are manually reviewed to eliminate misclassifications.

[3] SIP addresses are used for telephone extensions or VoIP systems.

In subsequent analysis, we group the product and organization types together due to their ambiguity, given that product names often include or imply their company's name.

Finally, information that cannot be classified using the above methodologies is marked as unidentified.

- **Unidentified**: The majority contain random strings. We perform further classification and analysis on the *unidentified* type in Section 6.3 (Table 9).

In summary, given the inherent ambiguity of human-readable strings and their classification challenges, we supplement automatic methods (i.e., regex matching and transformer models) with manual checks of certificate entries in CN, SAN, and issuer fields. For instance, using the pre-trained model [43], the precision (i.e., correctly identified personal names among those classified) and recall (i.e., correctly identified personal names among all actual names) for identifying personal names are both 0.9. Therefore, we perform manual checks to correct misclassifications and identify missed entries. This comprehensive approach ensures that the classification of each type of information is not only automated but also scrutinized to confirm accuracy.

6.1.2 Scope of Analysis. Here is the scope of our analysis regarding CN and SAN.

The SAN field. According to [4], several value types are defined that can be included in the SAN field, including domain names, IP addresses, email addresses, and URIs.[4] However, in our dataset, we observe a significant disparity in the utilization of these types: 99% of both IP address and URI types, as well as 99% of email address types, are left empty. When these types are used, they correctly match the corresponding information in SAN (e.g., IP addresses are presented when SAN is defined for the IP address type). In contrast, the SAN DNS type, while more frequently populated, often contain incorrect or non-standard values that are not domain names (details in Section 6.3). Given this disparity, our subsequent analysis will focus primarily on SAN DNS type within the SAN field. For simplicity, we will refer to the 'SAN DNS type' as 'SAN' unless otherwise specified.

Certificates from mutual TLS vs. non-mutual TLS. Most certificates used by non-mutual TLS connections are server certificates, and we observed that server certificates from mutual and non-mutual TLS connections exhibit similar trends. Therefore, we concentrate on certificates from mutual TLS, while the analysis of certificates from non-mutual TLS will be briefly discussed in Section 6.3.6. Furthermore, we approach our analysis from two perspectives: (1) *server* versus *client* certificates, and (2) certificates issued by *public CAs* versus *private CAs*.

6.2 Utilization of CN and SAN

Let us first examine how many certificates utilize CN and SAN. Table 7 presents the number of certificates with non-empty values in their CN or SAN (DNS) fields, along with their ratio to the total number of certificates.

Non-Empty	CN		SAN DNS	
	Number	%	Number	%
Server certs.	2,269,724	99.78	15,584	0.69
- Public CA	6,941	99.99	6,941	99.99
- Private CA	2,262,783	99.78	8,643	0.38
Client certs.	3,351,364	99.89	42,264	1.26
- Public CA	22,563	99.50	3,383	14.92
- Private CA	3,328,801	99.89	38,881	1.17

Table 7: Number of certificates (used by mutual TLS connections) with non-empty values in their CN or SAN fields. The ratio is calculated against the total number of certificates in each category.

Server vs. Client certificates. We first observe that CN is utilized more frequently compared to SAN. Specifically, about 99.8% of certificates, both server and client, contain values within their CN. However, the SAN fields show much lower utilization, where less than 1% of server certificates include values in SAN fields. This disparity is particularly notable considering that the use of CN is deprecated [6, 8, 38] (and the use of SAN is recommended). Furthermore, while clients typically do not have assigned domain names and are not expected to include SAN, a non-negligible number of client certificates have specified some information in their CN or SAN (further investigated in Section 6.3).

Public CA vs. Private CA certificates. It is apparent that certificates issued by public CAs tend to utilize SAN more than those issued by private CAs. This trend is more pronounced in server certificates compared to client certificates. The majority of non-empty SAN is accounted for by certificates issued by public CAs; in other words, most certificates issued by private CAs do not use SAN.

> **(*Takeaway*)** Despite the deprecation of CN, most certificates continue to utilize CN rather than SAN, and most certificates issued by private CAs do not utilize SAN.

6.3 What is in CN and SAN?

We now turn our attention to investigating the types of information in CN and SAN. Using the methodology and information types defined in Section 6.1, we categorize the (non-empty) entries contained within CN and SAN.

Table 8 shows the result, including the number and ratio of information types in each CN and SAN. Note that we exclude certificates shared by both server and client (discussed in Section 5.2) from this analysis, which will be analyzed separately in Section 6.3.5.

6.3.1 Server and Public CA. The majority of information types included in CN and SAN are domain names, as expected. All unidentifiable CNs are formatted as a combination of domain names, IP addresses, and some random strings, all of which have 'FNMT-RCM' as the issuer's organization name.

6.3.2 Server and Private CA. These certificates exhibit a greater variety of information types in both CN and SAN compared to those issued by public CAs. Additionally, there is a notable difference in the types of information used between CN and SAN. While most SANs include domain names, only 0.3% of CNs do so. Although not a large proportion, 6 CNs contain personal names. We also observe that a significant portion (79%) of CNs contain organization or product

[4] OpenSSL defines (general) types (e.g., 'GEN_DNS' for domain names, 'GEN_EMAIL' for emails addresses, and 'GEN_IPADD' for IP addresses) that can be used in the SAN field, each of which is parsed distinctly according to its specific type [34, 35].

Information type	Server certificates								Client certificates							
	Public CA				Private CA				Public CA				Private CA			
	CN		SAN		CN		SAN		CN		SAN		CN		SAN	
	Num.	%	Num.	%	Num.	%	Num.	%	Num.	%	Num.	%	Num.	%	Num.	%
Domain	6,867	99.94	6,871	100.00	7,438	0.34	7,562	87.69	3,153	14.11	3,165	99.94	6,278	0.19	7,723	19.88
IP	0	-	0	-	1,796	0.08	59	0.68	1	0.00	0	-	13	0.00	9	0.02
MAC	0	-	0	-	0	-	0	-	0	-	0	-	139	0.00	126	0.32
SIP	0	-	0	-	99,438	4.53	0	-	0	-	0	-	1,812	0.06	0	-
Email	0	-	0	-	0	-	0	-	2	0.01	0	-	945	0.03	24	0.06
User account	0	-	0	-	0	-	0	-	0	-	0	-	18,603	0.57	0	-
Personal name	0	-	0	-	6	0.00	0	-	133	0.59	0	-	43,539	1.33	4,901	12.62
Org/Product	0	-	0	-	1,741,333	79.30	681	7.90	5,660	25.33	1	0.03	3,017,852	92.49	5,561	14.32
Localhost	1	0.01	0	-	24	0.00	64	0.74	1	0.00	0	-	462	0.01	203	0.52
Unidentified	3	0.04	0	-	345,895	15.75	512	5.94	13,397	59.95	18	0.57	173,220	5.31	21,526	55.41

Table 8: Number (and ratio) of information types in CN and SAN, categorized by server and client certificates and certificates issued by public CAs and private CAs. Ratios smaller than 0.01 are denoted as 0.00. Note that since a SAN field in a certificate can contain multiple types of information, the cumulative percentage for SAN (in each column) may exceed 100%.

Unidentified type		Server	Client		
		Private CA	Public CA	Private CA	
		CN	CN	CN	SAN
Non-random		20%	-	16%	-
Random	by *Issuer*	1%	60%	30%	94%
	strlen = 8	46%	-	4%	-
	strlen = 32	17%	-	39%	-
	strlen = 36	9%	40%	2%	1%

Table 9: Detailed classification of unidentified types; non-random and random strings.

names; 88% of these use 'WebRTC' as their CN. Excluding this case, half of the remaining CNs list 'twilio' and 29% list 'hangouts'.

Unidentified type. We notice a significantly higher number of unidentified type. Therefore, we conduct further investigations by examining the issuer field or formats of the strings to identify random v.s. non-random strings. The random strings are then categorized based on recognizable features such as the issuer field (in the certificates) or the string lengths (8, 32, and 36). Table 9 shows the result. The majority (80%) of the strings used in the CN field (of server certificates issued by private CAs) appear to be random, such as hash values. Among them, 9% are 36 characters long, formatted as Universally Unique Identifier (UUID); such UUIDs could potentially be used for user identification if combined with other techniques. In SAN, 57% of unidentified strings are formatted as a combination of the CN string, the string 'TLS', and some random numbers. However, we do not observe any other commonalities among them, and most of these certificates have different issuers.

6.3.3 *Client and Public CA.*

99.9% of certificates contain domain names in SAN, whereas only 14.1% of CNs contain domain names. Interestingly, we typically expect that clients are not assigned domain names. However, in our dataset, about 14% of client certificates include domain names listed in their CN. Further investigation reveals that 38% of these domain names are associated with email services (by including keywords like 'smtp', 'mx', 'mta', or 'mail' in their domain names), and 24% are linked to Cisco's Webex service. Additionally, 133 certificates include personal names which are issued by various issuers. 25% of CNs list organization or product names, with 99% identifying 'Hybrid Runbook Worker' as a CN, a feature associated with Microsoft Azure.

Unidentified type. 59.95% of CNs in client certificates issued by public CAs are classified as unidentified. Upon manual inspection, we notice that almost all of these entries comprise random strings. However, some patterns emerge upon examining their issuers: 46% of these unidentified types list 'Microsoft Azure Sphere *w/ Random string*' as the issuer CN, and 10% are in UUID formats associated with Apple ('Apple iPhone Device CA' as the issuer CN). For the remaining entries, most are in UUID formats (e.g., the CN string length is 36) but lack any specific information in their issuer fields.

6.3.4 *Client and Private CA.*

Client certificates issued by private CAs display the most diverse types of information, predominantly featuring private information, including 18,603 user accounts and 43,539 personal names in CN. All user accounts are IDs used for accessing campus services such as websites (or other campus-related services). Of the certificates listing personal names, 93% are issued by campus CAs, while 7% are issued by other private CAs. This highlights the typical use of client certificates for user authentication, frequently including detailed information about the users. The organization or product type is prominently represented, with 98.7% associated with 'WebRTC'; among the remaining certificates, 22% are related to 'Lenovo' products, and 16% are connected to the 'Android Keystore' system.

Unidentified type. 16% of unidentified CNs are non-random strings, but often with meaningless formats such as "__transfer__" or "Dtls"; 14% of these are associated with a file transfer service. For CNs with random strings, an examination of the issuer fields reveals that 22% are related to services provided by AT&T, Red Hat, or Samsung. In SAN, 94% of the random strings can be recognized by their issuer fields, with the majority being issued by campus CAs.

6.3.5 *Information in the shared certificates.*

For certificates shared by both server and client (Section 5.2), we identified a total of 67,221 certificates (from mutual TLS connections). The majority (99.7%) are issued by private CAs. Next, among the 67,221 certificates, 98.4% have values in their CN fields, while only 0.4% have values in their SAN fields. Detailed breakdowns are in Table 13 (in Appendix D). The overall trend is similar to that of server certificates (in Table 7 and Table 8).

Consistent with server certificate results, certificates issued by public CAs exclusively contain domain names in both CN and SAN fields. In contrast, private CA-issued certificates show more diversity: 11% include organization or product names, with 64.1% using 'WebRTC' and 27.6% using 'hangouts' as CNs. Additionally, 85%

contain unidentified information, 84.3% of which are non-random strings related to file transfer services. Of the random strings, 81.6% are 8-character hash values.

6.3.6 Information in the certificates from non-mutual TLS. We also examine certificates from non-mutual TLS connections. Given that most non-mutual TLS connections utilize only server certificates, we focus on the information contained in server certificates.

First, these certificates are predominantly issued by public CAs (85%), in contrast to the mutual TLS case, where 99% are issued by private CAs. Next, for server certificates issued by public CAs, 99% have values in both their CN and SAN fields, consistent with the mutual TLS case. However, server certificates issued by private CAs are more likely to have values in their SAN fields: 10.5% compared to 0.4% in the mutual TLS case (see Table 14a, Appendix D).

The types of information in these server certificates from non-mutual TLS are similar to those in mutual TLS connections. Although fewer in number compared to mutual TLS, certificates (from non-mutual TLS) issued by private CAs also contain private information, such as user accounts or personal names (see Table 14b, Appendix D). Unlike mutual TLS, server certificates from private CAs have a higher ratio of unidentified types (26.7% vs. 5.9%), with 39% of these being non-random strings like 'hmpp' or 'Dtls'.

6.3.7 Summary. Our analysis identifies a diverse range of information types included in both CN and SAN. Specifically, in SAN, despite explicit types being available for Email and IP addresses, numerous certificates include these within the SAN DNS field. Additionally, we observe notable differences between client and server certificates, as well as between certificates issued by public CAs and private CAs. Server certificates from public CAs conform to formatting standards by including domain names in their SAN (and also CN). However, client certificates, even those from public CAs, tend to encompass a broader variety of information types. The inclusion of diverse information types is more pronounced in certificates issued by private CAs.

Furthermore, we observe that many certificates from private CAs include sensitive information, such as user accounts or personal names; this trend is mainly observed in client certificates but is also present (albeit to a lesser extent), in server certificates issued by private CAs. Considering that certificates are sent unencrypted during the TLS handshake (in versions before TLS 1.3 [39]), the presence of private information in certificates could pose significant privacy risks, especially for server certificates.

(*Takeaway*) We identify various types of information in CN and SAN, many of which do not conform to formatting standards. In particular, many certificates from private CAs include sensitive information, posing privacy concerns.

7 Discussion

Mutual TLS prevalence. Our study reveals a modest increase in usage of mutual TLS within a campus network, rising from 1.99% to 3.61% of total TLS connections over the 23-month period. However, the overall prevalence of TLS remains relatively low. Several factors may contribute to this. For certain applications, especially those serving a broad user base, implementing client authentication can complicate the user experience. Additionally, many organizations may prefer alternative security models such as API keys and OAuth, particularly if these methods are perceived as adequate for their needs, reducing the incentive to adopt mutual TLS. Moreover, the introduction of client authentication presents challenges in certificate management, including issues related to revocation and renewal, which can add operational overhead. Individual devices typically pose greater security risks than centralized servers, which can elevate the likelihood of client-side private key compromise if not managed appropriately.

Enhancing privacy of client certificates. To mitigate privacy concerns associated with client certificates in mutual TLS, several strategies can be adopted. Primarily, client certificates should contain only the essential information required for authentication, excluding Personally Identifiable Information (PII) or other sensitive data. Additionally, secure storage practices, such as employing Hardware Security Modules (HSMs), are critical to prevent unauthorized access. Robust certificate management practices are also vital for minimizing the risk of privacy breaches. Moreover, conducting regular privacy impact assessments and adversarial testing can effectively address potential vulnerabilities and leakage issues.

Limitations of our approach. Although our study highlights some concerning practices of client authentication, our passive monitoring approach limits our ability to assess the specific implementations of client authentication mechanisms, which can differ significantly across various software and applications. A potential avenue for future research could involve conducting code-level investigations and adversarial testing to gain a deeper understanding of these implementations.

8 Conclusion

In this paper, we conducted an in-depth measurement study using TLS connection data from a campus network over 23 months to understand the real-world usage of certificates associated with mutual TLS and the information revealed in the CN and SAN fields. We found an increasing trend in the utilization of mutual TLS. However, we identified several problematic practices, such as the lack of a valid client issuer, the use of client certificates with dummy serial numbers, the use of identical certificates by both endpoints (i.e., a server and a client), and the continuous use of expired client certificates. Our analysis also revealed various types of information in the CN and SAN fields, including sensitive details, such as personal names, which raises privacy concerns. Overall, our study contributes to a better understanding of mutual TLS usage and associated privacy implications, providing a basis for future research and improvements in secure practices.

Acknowledgments

We thank anonymous reviewers for their insightful and constructive suggestions and feedback. This work is supported by National Science Foundation CNS-2154962 and CNS-2319421, and the Commonwealth Cyber Initiative.

References

[1] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, et al. 2019. Let's Encrypt: an automated certificate authority to encrypt

the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2473–2487.

[2] Apple. 2024. Available trusted root certificates for Apple operating systems. https://support.apple.com/en-us/103272 (accessed Aug 26, 2024).

[3] BigPicture. 2023. 17M+ Company Dataset - BigPicture 2023 Q4 Free Company Dataset. https://www.kaggle.com/datasets/mfrye0/bigpicture-company-dataset (accessed Aug 26, 2024).

[4] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and David Cooper. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280. https://doi.org/10.17487/RFC5280

[5] Chad Brubaker, Suman Jana, Baishakhi Ray, Sarfraz Khurshid, and Vitaly Shmatikov. 2014. Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In *2014 IEEE Symposium on Security and Privacy*. IEEE, 114–129.

[6] CA/Browser Forum. 2024. Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates Version 2.0.4. https://cabforum.org/working-groups/server/baseline-requirements/documents/TLSBRv2.0.4.pdf (accessed Aug 26, 2024).

[7] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and analysis of private key sharing in the https ecosystem. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 628–640.

[8] Chrome Platform Status. 2022. Feature: Support for commonName matching in Certificates (Removed). https://chromestatus.com/feature/4981025180483584 (accessed Aug 26, 2024).

[9] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. 2016. Measuring and applying invalid SSL certificates: The silent majority. In *Proceedings of the 2016 Internet Measurement Conference*. 527–541.

[10] CloudFlare. 2024. What is mutual TLS (mTLS). https://www.cloudflare.com/learning/access-management/what-is-mutual-tls/ (accessed Aug 26, 2024).

[11] crt.sh. 2015. Certificate Transparency Logs. https://crt.sh/ (accessed Aug 26, 2024).

[12] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. 2013. Analysis of the HTTPS certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference*. 291–304.

[13] Elaine Barker, Quynh Dang. 2015. Recommendation for Key Management - Part 3: Application-Specific Key Management Guidance. https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-57pt3r1.pdf (accessed Aug 26, 2024).

[14] Syed Muhammad Farhan and Taejoong Chung. 2023. Exploring the Evolution of TLS Certificates. In *International Conference on Passive and Active Network Measurement*. Springer, 71–84.

[15] FileWave. 2024. FileWave. https://www.filewave.com/ (accessed Aug 26, 2024).

[16] Lucas Foppe, Jeremy Martin, Travis Mayberry, Erik C Rye, and Lamont Brown. 2018. Exploiting tls client authentication for widespread user tracking. *Proceedings on Privacy Enhancing Technologies* (2018).

[17] The Internet Corporation for Assigned Names and Numbers (ICANN). 2024. Internet Assigned Numbers Authority (IANA). https://www.iana.org/ (accessed Aug 26, 2024).

[18] Python Software Foundation. 2024. ipaddress - IPv4/IPv6 manipulation library. https://docs.python.org/3/library/ipaddress.html (accessed Aug 26, 2024).

[19] The Linux Foundation. 2024. Common CA Database. https://www.ccadb.org/ (accessed Aug 26, 2024).

[20] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. 2012. The most dangerous code in the world: validating SSL certificates in non-browser software. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 38–49.

[21] Sina Keshvadi and Yogesh Sharma. 2023. Exploring HTTPS Certificate Ecosystem: Analyzing the Entire IPv4 Address Space. In *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 547–552.

[22] Doowon Kim, Haehyun Cho, Yonghwi Kwon, Adam Doupé, Sooel Son, Gail-Joon Ahn, and Tudor Dumitras. 2021. Security analysis on practices of certificate authorities in the HTTPS phishing ecosystem. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. 407–420.

[23] Dr. John C. Klensin. 2008. Simple Mail Transfer Protocol. RFC 5321. https://doi.org/10.17487/RFC5321

[24] John Kurkowski. 2024. tldextract. https://github.com/john-kurkowski/tldextract (accessed Aug 26, 2024).

[25] Hugo Landau. 2023. Client certificates aren't universally more secure. https://www.devever.net/~hl/clientcert, (accessed Aug 26, 2024).

[26] Let's Encrypt. 2024. Let's Encrypt: ACME Client Implementations. https://letsencrypt.org/docs/client-options/ (accessed Aug 26, 2024).

[27] Linkerd. 2024. Automatic mTLS. https://linkerd.io/2.15/features/automatic-mtls/, (accessed Aug 26, 2024).

[28] Microsoft. 2024. Release notes - Microsoft Trusted Root Certificate Program. https://learn.microsoft.com/en-us/security/trusted-root/release-notes (accessed Aug 26, 2024).

[29] Mozilla Foundation. 2022. Public Suffix List. https://publicsuffix.org/ (accessed Aug 26, 2024).

[30] Mozilla Wiki. 2024. Mozilla's CA Certificate Program. https://wiki.mozilla.org/CA (accessed Aug 26, 2024).

[31] MQTT. 2022. MQTT: The Standard for IoT Messaging. https://mqtt.org/ (accessed Aug 26, 2024).

[32] The University of Chicago. 2024. Globus Compute. https://www.globus.org/ (accessed Aug 26, 2024).

[33] OpenSSL. 2024. OpenSSL: Cryptography and SSL/TLS Toolkit. https://www.openssl.org/ (accessed Aug 26, 2024).

[34] OpenSSL. 2024. openssl/crypto/x509v3/v3_sanċ. https://github.com/openssl/openssl/blob/fa338aa7cd1e893679c3e1c47465dcb11f90abfb/crypto/x509/v3_san.c, (accessed Aug 26, 2024).

[35] OpenSSL. 2024. openssl/include/openssl/x509v3.h.in. https://github.com/openssl/openssl/blob/fa338aa7cd1e893679c3e1c47465dcb11f90abfb/include/openssl/x509v3.h.in, (accessed Aug 26, 2024).

[36] Arnis Parsovs. 2013. Practical issues with TLS client certificate authentication. *Cryptology ePrint Archive* (2013).

[37] People Data Labs. 2019. 7+ Million Company Dataset - People Data Labs 2019 Global Company Dataset. https://www.kaggle.com/datasets/peopledatalabssf/free-7-million-company-dataset (accessed Aug 26, 2024).

[38] Eric Rescorla. 2000. HTTP Over TLS. RFC 2818. https://doi.org/10.17487/RFC2818

[39] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. https://doi.org/10.17487/RFC8446

[40] Peter Saint-Andre and Jeff Hodges. 2011. Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS). RFC 6125. https://doi.org/10.17487/RFC6125

[41] Jim Sermersheim. 2006. Lightweight Directory Access Protocol (LDAP): The Protocol. RFC 4511. https://doi.org/10.17487/RFC4511

[42] Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen K Lenstra, David Molnar, Dag Arne Osvik, and Benne de Weger. 2008. MD5 considered harmful today, creating a rogue CA certificate. In *25th Annual Chaos Communication Congress*.

[43] spaCy. 2024. Trained Pipelines: English. https://spacy.io/models/en (accessed Aug 26, 2024).

[44] Matthias Wachs, Quirin Scheitle, and Georg Carle. 2017. Push away your privacy: Precise user tracking based on tls client certificate authentication. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–9.

[45] Wei Xia, Mingxin Cui, Wei Wang, Yangyang Guan, Zhenzhen Li, Zhen Li, and Gang Xiong. 2021. Illuminate the shadow: A comprehensive study of tls client certificate ecosystem in the wild. In *2021 28th International Conference on Telecommunications (ICT)*. IEEE, 1–5.

[46] Wei Xia, Wei Wang, Xin He, Gang Xiong, Gaopeng Gou, Zhenzhen Li, and Zhen Li. 2021. Old habits die hard: A sober look at tls client certificates in the real world. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 83–90.

[47] Wei Xia, Qiyu Zhang, Xin He, Wei Wang, Zhen Li, and Gang Xiong. 2021. After everything is connected: A client certificate-oriented perspective of iot device security analysis. In *Proceedings of the 2021 9th International Conference on Information Technology: IoT and Smart City*. 321–325.

[48] Zuyong Yin, Qi Zhou, Junqiu Qu, and Fanrong Lv. 2023. How Far is User Privacy Leakage: A Revisit of Client Certificate Usage. In *2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*. IEEE, 279–284.

[49] Zeek. 2023. An Open Source Network Security Monitoring Tool. https://zeek.org/ (accessed Aug 26, 2024).

[50] Zeek. 2023. Zeek Dynamic Protocol Detection. https://docs.zeek.org/en/master/logs/dpd.html (accessed Aug 26, 2024).

A Ethics

The data logging procedure of this research has been closely supervised by the university's InfoSec department and has been ethically approved by the Institutional Review Board (IRB). This study is restricted to specifically authorized fields within the processed Zeek logs, with no exposure to the raw network traffic. All data storage and computation activities associate with this study are securely conducted within the university's protected cluster, which is only accessible through a restricted network. Furthermore, access to the data is limited to selected personnel who has completed the university's comprehensive security training. This effectively mitigates the risk of potential data leakage and ensures the protection of sensitive and private information.

	SLD	Client certificate issuer organization	Server certificate issuer organization	#. involved clients	Duration of activity (days)
Out	fireboard.io	Internet Widgits Pty Ltd	Internet Widgits Pty Ltd	9	618
	amazonaws.com	Internet Widgits Pty Ltd	Internet Widgits Pty Ltd	7	17
	- (missing SNI)	Internet Widgits Pty Ltd	Internet Widgits Pty Ltd	1	1

Table 10: Details on certificates with dummy issuers presented by both the client and the server in mutual TLS connections.

	SLD	Incorrect dates in	Issuer	Certificate validity period (not valid before, not valid after)	#. involved clients	Duration of activity (days)
In	- (missing SNI)	Client certificate	rcgen	(1975, 1757)	2	42
Out	idrive.com	Client certificate	IDrive Inc Certificate Authority	(2019, 1849)	2,887	701
		Server certificate	IDrive Inc Certificate Authority	(2020, 1850)	718	701
	clouddevice.io	Client certificate	Honeywell International Inc	(2021, 1815)	1599	701
				(2023, 1815)	46	258
	alarmnet.com	Client certificate	Honeywell International Inc	(2021, 1815)	1864	696
				(2023, 1815)	70	252
	- (missing SNI)	Client certificate	SDS	(1970, 1831)	17	474
		Server certificate	SDS	(1970, 1831)	17	474
	ayoba.me	Client certificate	OpenPGP to X.509 Bridge	(2022, 2022)*	15	147
	ibackup.com	Client certificate	IDrive Inc Certificate Authority	(2019, 1849)	4	311
	crestron.io	Client certificate	Crestron Electronics Inc	(2020, 1816)	3	1
	- (missing SNI)	Server certificate	media-server	(2157, 2023)	2	106
	- (missing SNI)	Client certificate	IceLink	(2048, 1996)	1	1

* Identical timestamps.

Table 11: Details on certificates with incorrect dates used in mutual TLS.

B Certificates with dummy issuers

Table 10 presents details on certificates with dummy issuers utilized by both the client and the server. These certificates, regardless of whether they are server-side or client-side, are all issued by the same entity, 'Internet Widgits Pty Ltd', which is the default issuer name in OpenSSL for certificate creation. This practice raises concerns about the integrity of the certificate validation process at both ends and undermines the security framework reliant on trusted CAs.

C Certificates with incorrect dates

This section outlines our observations regarding certificates used in mutual TLS that exhibit incorrect date configurations.

Table 11 and Table 12 show details on certificates with incorrect dates observed in connections. All these certificates exhibit a 'not valid before' date that follows the 'not valid after' date, affecting thousands of clients. Specifically, domains such as idrive.com use such certificates at both ends. This configuration practice suggests a mishandling of the certificate validation process, which, if left unaddressed, renders connections susceptible to MITM attacks.

D Information in certificates

Table 13 presents the statistics of certificates shared by both endpoints. Specifically, Table 13a shows the number of shared certificates with values in their CN and SAN fields, while Table 13b outlines the types of information contained in those certificates.

Table 14 shows the statistics for certificates used in non-mutual TLS connections; note that we only consider server certificates used for server authentication. Table 14a displays the number of certificates (from non-mutual TLS connections) with values in their CN and SAN fields, while Table 14b details the types of information in those certificates.

SLD	Client certificate issuer	Client certificate (not before, not after)	Server certificate issuer	Server certificate (not before, not after)	#. involved clients	Duration of activity (days)
idrive.com	IDrive Inc Certificate Authority	(2019-08-02, 1849-10-24)	IDrive Inc Certificate Authority	(2020-07-03, 1850-09-25)	718	701
- (missing SNI)	SDS	(1970-01-01, 1831-11-22)	SDS	(1970-01-01, 1831-11-22)	17	474

Table 12: Details on certificates with incorrect dates employed by both the client and the server in mutual TLS connections.

Non-Empty	CN		SAN DNS	
	Number	%	Number	%
Certificates	66,154	98.41	251	0.37
- Public CA	216	100.00	216	100.00
- Private CA	65,938	98.41	35	0.05

(a) Number of shared certificates used by both server and client, with non-empty values in their CN or SAN fields. The ratio is calculated against the total number of certificates in each category.

Information type	Public CA				Private CA			
	CN		SAN		CN		SAN	
	Num.	%	Num.	%	Num.	%	Num.	%
Domain	216	100.00	216	100.00	65	0.10	27	77.14
IP	0	-	0	-	210	0.32	7	20.00
MAC	0	-	0	-	0	-	0	-
SIP	0	-	0	-	1,838	2.79	0	-
Email	0	-	0	-	0	-	0	-
User account	0	-	0	-	0	-	0	-
Personal name	0	-	0	-	3	0.00	0	-
Org/Product	0	-	0	-	7,849	11.90	0	-
Localhost	0	-	0	-	8	0.01	8	22.86
Unidentified	0	-	0	-	55,965	84.88	0	-

(b) Number (and ratio) of information types in shared certificates. Ratios smaller than 0.01 are denoted as 0.00. Note that since a SAN field in a certificate can contain multiple types of information, the cumulative percentage for SAN (in each column) may exceed 100%.

Table 13: Information in the CN and SAN fields of certificates shared by both server and client.

Non-Empty	CN		SAN DNS	
	Number	%	Number	%
Certificates	3,706,858	99.95	3,225,308	86.96
- Public CA	3,168,196	99.98	3,168,361	99.99
- Private CA	538,662	99.72	56,947	10.54

(a) Number of certificates from non-mutual TLS connections, with non-empty values in their CN or SAN fields. The ratio is calculated against the total number of certificates in each category.

Information type	Public CA				Private CA			
	CN		SAN		CN		SAN	
	Num.	%	Num.	%	Num.	%	Num.	%
Domain	3,167,630	99.98	3,168,357	100.00	71,506	13.27	40,980	71.96
IP	380	0.12	2	0.00	2,678	0.50	719	1.26
MAC	0	-	0	-	9	0.00	1	0.00
SIP	0	-	0	-	6,506	1.21	0	-
Email	0	-	0	-	10	0.00	0	-
User account	0	-	0	-	192	0.04	0	-
Personal name	3	0.00	0	-	588	0.11	66	0.12
Org/Product	3	0.00	21	0.00	396,238	73.56	1,424	2.50
Localhost	1	0.00	0	-	1,585	0.29	608	1.07
Unidentified	179	0.06	50	0,00	59,350	11.02	15,227	26.74

(b) Number (and ratio) of information types in certificates from non-mutual TLS connections. Ratios smaller than 0.01 are denoted as 0.00. Note that since a SAN field in a certificate can contain multiple types of information, the cumulative percentage for SAN (in each column) may exceed 100%.

Table 14: Information in the CN and SAN fields of certificates from non-mutual TLS connections.