**HOCHSCHULE für Angewandte Wissenschaften LANDSHUT**

**UNIVERSITY of Applied Sciences**

**FAKULTÄT Informatik**

**HOCHSCHULE LANDSHUT**

FAKULTÄT INFORMATIK

**Echtzeitbasierte Netzwerkdatenanalyse zur Ermittlung verwendeter Root Zertifikate**

**Masterarbeit**

vorgelegt von

**Sebastian Bilda**

aus Landshut

eingereicht am: .......................

Betreuer: Johann Uhrmann

# Erklärung zur Masterarbeit

Name, Vorname   
der/des Studierenden: …………………………………………….

Hiermit erkläre ich, dass ich die Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Landshut, den ……………. …..………………………………….  
 (Unterschrift der/des Studierenden)

Abstract

Inhaltsverzeichnis

[Erklärung zur Masterarbeit - 4 -](#_Toc65828847)

[Abstract - 6 -](#_Toc65828848)

[Inhaltsverzeichnis - 7 -](#_Toc65828849)

[1 Einleitung - 8 -](#_Toc65828850)

[1.1 Problemstellung und Motivation - 8 -](#_Toc65828851)

[1.2 Zielsetzung - 10 -](#_Toc65828852)

[2 Grundlagen - 11 -](#_Toc65828853)

[2.1 Public Key Infrastructure - 11 -](#_Toc65828854)

[2.2 Zertifizierungsstellen - 11 -](#_Toc65828855)

[2.3 Digitale Zertifikate - 11 -](#_Toc65828856)

[2.3.1 X.509 Zertifikate - 12 -](#_Toc65828857)

[2.3.2 Arten von Zertifikaten - 12 -](#_Toc65828858)

[2.3.3 Aufbau und Inhalt - 12 -](#_Toc65828859)

[2.4 Transport Layer Security - 12 -](#_Toc65828860)

[2.4.1 TLS Handshake - 13 -](#_Toc65828861)

[2.4.2 TLS Record - 14 -](#_Toc65828862)

[2.5 Windows Certificate Trust Store - 14 -](#_Toc65828863)

[3 Hauptteil - 15 -](#_Toc65828864)

[3.1 Stand der Forschung - 15 -](#_Toc65828865)

[3.2 Methodik/Vorgehensweise - 16 -](#_Toc65828866)

[3.3 Implementierung - 18 -](#_Toc65828867)

[3.3.1 Anforderungsanalyse - 18 -](#_Toc65828868)

[3.3.2 Aufbau und Struktur - 19 -](#_Toc65828869)

[3.3.3 Probleme bei der Implementierung? - 20 -](#_Toc65828870)

[3.4 Ergebnisse - 20 -](#_Toc65828871)

[4 Fazit - 23 -](#_Toc65828872)

[4.1 Zusammenfassung - 23 -](#_Toc65828873)

[4.2 Reflexion - 23 -](#_Toc65828874)

[4.3 Ausblick - 23 -](#_Toc65828875)

[Literaturverzeichnis - 24 -](#_Toc65828876)

[Abbildungsverzeichnis - 25 -](#_Toc65828877)

[Listingverzeichnis - 26 -](#_Toc65828878)

[Anhang - 27 -](#_Toc65828879)

# Einleitung

## Problemstellung und Motivation

Jede Person muss sich selbst mittels eines Personalausweises identifizieren können. So eine ähnliche Funktion besitzen digitale Zertifikate in der Online-Welt. Die Authentizität des Kommunikationspartners wird mit einer Prüfung seines Zertifikats sichergestellt. Diese Zertifikate werden von Zertifizierungsstellen (CAs) ausgestellt, denen der Nutzer (Client, Browser, Betriebssystem) vertraut. Zertifikate sind aus dem Internet nicht mehr wegzudenken. Sie erhöhen die Sicherheit beim „surfen“, identifizieren ihren Besitzer, enthalten allgemeine Informationen über diesen und von größter Bedeutung, sie ermöglichen es zusammen mit der Public Key Infrastruktur (PKI), Informationen im WWW sicher und verschlüsselt zu übertragen. Verschlüsselte Verbindungen werden in der Regel durch TLS-basierte Authentifizierung von mindestens einem Kommunikationspartner abgesichert. Die Zertifikate werden von sogenannten Zertifizierungsstellen (Root Certificate Authority) ausgestellt, denen jeder Browser und jedes System „vertraut“. Aus jüngsten Ereignissen, ist jedoch bekannt, dass ein Unternehmen, dass auch eine TLS-Zertifizierungsstelle betreibt, im staatlichen Auftrag Menschenrechtsaktivisten durch Sicherheitslücken angreift.

Somit kann aus diesem Zusammenhang die Menge an vordefinierten Zertifizierungsstellen in Systemen oder Browsern in Frage gestellt werden. An dem Beispiel von dem Unternehmen Dark Matter kann das Problem spezifiziert werden, dass durch eine Phishing Mail auf eine Website leitet auf der Malicious Code heruntergeladen wird, diese Website wurde zum Beispiel von einer CA signiert die mit Hackingaktivitäten in Verbindung steht, d. h. mein System vertraut dieser Website und keine Meldung über die Sicherheit ihrer Daten kommt. Zusätzlich können MiM-Angriffe leichter vorgenommen werden, und der Angreifer kann die gesamte Kommunikation mitlesen.

Es gibt verschiedene Ansätze diesem Problem entgegenzuwirken. Zum einen eine Initiative von Google namens Certificate Transparenzy, die durch eine zusätzliche öffentliche Instanz, auf der alle neu registrierten Zertifkate in einem kryptografischen Verfahren gespeichert sind, um ein erneutes Ausstellen eines Zertifikates für eine Domain zu verhindern. Dieses Verfahren wird im Abschnitt Stand der Forschung nochmal genauer erklärt. Zum anderen gibt es die Möglichkeit über den CAA Record die Zertifkate genauer zu überprüfen.

Oder, was in dieser Masterarbeit untersucht werden soll, man versucht durch eine Einschränkung der Zertifizierungsstellen das Surfen im Internet noch sicherer zu machen.

Es gibt derzeit keinen etablierten Mechanismus, um dieses Vertrauen feingranularer zu konfigurieren oder überhaupt zu verifizieren, ob dieses Vertrauen notwendig bzw. gerechtfertigt ist. Natürlich ist es möglich diese Liste manuell anzupassen, jedoch bringt dies viel Aufwand mit sich und nach welchen Kriterien werden die Zertifizierungsstellen bewertet? Jede Person muss sich selbst mittels eines Personalausweises identifizieren können. So eine ähnliche Funktion besitzen digitale Zertifikate in der Online-Welt. Die Authentizität des Kommunikationspartners wird mit einer Prüfung seines Zertifikats sichergestellt. Diese Zertifikate werden von Zertifizierungsstellen (CAs) ausgestellt, denen der Nutzer (Client, Browser, Betriebssystem) vertraut. Zertifikate sind aus dem Internet nicht mehr wegzudenken. Sie erhöhen die Sicherheit beim „surfen“, identifizieren ihren Besitzer, enthalten allgemeine Informationen über diesen und von größter Bedeutung, sie ermöglichen es zusammen mit der Public Key Infrastruktur (PKI), Informationen im WWW sicher und verschlüsselt zu übertragen. Verschlüsselte Verbindungen werden in der Regel durch TLS-basierte Authentifizierung von mindestens einem Kommunikationspartner abgesichert. Die Zertifikate werden von sogenannten Zertifizierungsstellen (Root Certificate Authority) ausgestellt, denen jeder Browser und jedes System „vertraut“. Aus jüngsten Ereignissen, ist jedoch bekannt, dass ein Unternehmen, dass auch eine TLS-Zertifizierungsstelle betreibt, im staatlichen Auftrag Menschenrechtsaktivisten durch Sicherheitslücken angreift.

Somit kann aus diesem Zusammenhang die Menge an vordefinierten Zertifizierungsstellen in Systemen oder Browsern in Frage gestellt werden. An dem Beispiel von dem Unternehmen Dark Matter kann das Problem spezifiziert werden, dass durch eine Phishing Mail auf eine Website leitet auf der Malicious Code heruntergeladen wird, diese Website wurde zum Beispiel von einer CA signiert die mit Hackingaktivitäten in Verbindung steht, d. h. mein System vertraut dieser Website und keine Meldung über die Sicherheit ihrer Daten kommt. Zusätzlich können MiM-Angriffe leichter vorgenommen werden, und der Angreifer kann die gesamte Kommunikation mitlesen.

Mißbrauch mit ca certificaten

<https://resources.sei.cmu.edu/asset_files/WhitePaper/2001_019_001_496192.pdf>

Beschreibung eines Vorfalls bei dem VeriSign signierte Zertifikate an unberechtigte Personen ausgestellt hat die sich als Microsoft Mitarbeiter ausgegeben haben und somit Programmcode damit signieren konnten, um diesen dann unter Windows ohne Probleme auszuführen.

Oder, was in dieser Masterarbeit untersucht werden soll, man versucht durch eine Einschränkung der Zertifizierungsstellen das Surfen im Internet noch sicherer zu machen.

Es gibt derzeit keinen etablierten Mechanismus, um dieses Vertrauen feingranularer zu konfigurieren oder überhaupt zu verifizieren, ob dieses Vertrauen notwendig bzw. gerechtfertigt ist. Natürlich ist es möglich diese Liste manuell anzupassen, jedoch bringt dies viel Aufwand mit sich und nach welchen Kriterien werden die Zertifizierungsstellen bewertet?

## Zielsetzung

Aufgrund der oben genannten Problemstellung soll deshalb hier ein Analyse Werkzeug programmiert werden, das anhand von einem Netzwerkverkehr Mitschnitt diese Zertifizierungsstellen extrahiert und dann in den jeweiligen Systemen automatisiert angepasst wird. Des Weiteren soll anhand der Netzwerkverkehrs Auswertungen erstellt werden, die die Anzahl der ausgestellten Zertifikate einer Certificate Authorities (CA) graphisch darstellt. Auf welche Zeit konvergiert die Anzahl der tatsächlich verwendeten CA’s? Auswertung der Häufigkeit einer CA in Bezug auf einen Haushalt. Auswertung welche CA’s verwendet werden. Diese Statistiken sollen eine Basis dafür bilden, um die Anzahl der Zertifizierungsstellen in den Systemen einzuschränken.

# Grundlagen

In diesem Kapitel sollen die Grundlagen für das Verständnis der Arbeit beschrieben werden und die Schlüsselbegriffe definiert. Wie in der Zielsetzung oben beschrieben geht es um Netzwerke, dem SSL/TLS Protokoll, digitale Zertifikate und Zertifizierungsstellen. Diese Themengebiete sollen nun grundlegend erläutert werden.

Vorerst soll geklärt werden was eine PKI (Public Key Infrastructure) ist und wofür diese da ist. Weitergehend werden Zertifikate, Arten von Zertifikaten und deren Aufbau und Inhalt erläutert.

## Public Key Infrastructure

Eine Public Key Infrastructure (PKI) oder auf Deutsch übersetzt – *Infrastruktur für öffentliche Schlüssel* – ist "das gesamte Konstrukt rund um die Absicherung der Datenkommunikation und die Identitätskontrolle im Netzwerk mithilfe von Zertifikaten". [[3]](#_CTVL001db152e864b5f482587a8c80e6c73c7c7) Mit Hilfe einer PKI wird die Sicherheit der Kommunikation durch Authentifizierung, Datenverschlüsselung und digitale Signaturen erhöht. Grundsätzlich beruhen die Dienste einer PKI auf der ordnungsgemäßen Verwendung von öffentlichen/privaten Schlüsselpaaren. Die öffentliche Komponente dieses Schlüsselpaars wird in Form eines Public-Key-Zertifikats ausgestellt und kann in Verbindung mit dem entsprechenden Algorithmus zur Verifizierung einer digitalen Signatur, zur Verschlüsselung von Daten oder zu beidem verwendet werden. [[1]](#_CTVL00153b2fcd3a0084b6c938e85d9d05fb933)

## Zertifizierungsstellen

Zertifizierungsstellen [[3]](#_CTVL001db152e864b5f482587a8c80e6c73c7c7)

Zertifizierungsstellen stellen digitale Zertifikate aus <https://www.globalsign.com/de-de/ssl-information-center/zertifizierungsstellen-vertrauenshierarchien>

## Digitale Zertifikate

Beginnen möchte ich mit dem Begriff Zertifikat, der nach Peter Kleop folgendermaßen definiert ist:

“Ein digitales Zertifikat bindet einen öffentlichen Schlüssel an eine Entität (Benutzer, Organisation, Computer) und beinhaltet zusätzliche Informationen, wie Sperrlisteninformationen und vieles mehr." [[3]](#_CTVL001db152e864b5f482587a8c80e6c73c7c7)

### X.509 Zertifikate

<https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02103/BSI-TR-02103.pdf?__blob=publicationFile&v=4>

### Arten von Zertifikaten

Root, Intermediate, SSL

Verwendungszweck von Zertifikaten

### Aufbau und Inhalt

Technisch in detail? Wie werden Zertifikate erstellt? Formel? Was beinhalten Sie? CRL? Aufbau von X.509 nach RFC

Self Signed??

## Transport Layer Security

TLS (Transport Layer Security) ist der Nachfolger von SSL (Secure Sockets Layer), dabei handelt es sich um einen Standard. In der aktuellen Version 1.3 hat TLS das primäre Ziel eine Verbindung zwischen zwei kommunizierenden Anwendungen zu gewährleisten. Die einzige Anforderung an die darunterliegende Transportschicht ist ein zuverlässiger, ordnungsgemäßer Datenstrom. Grundsätzlich besteht TLS aus zwei Komponenten (<https://tools.ietf.org/html/rfc8446#section-1>):

* TLS Handshake Protocol (2.4.1)
* TLS Record Protocol (2.4.2)

TLS ist unabhängig vom Anwendungsprotokoll, d. h. die Protokolle höherer Ebenen können transparent darauf aufsetzten, zum Beispiel HTTP das TLS verwendet wird zu HTTPS. Der Standard beschreib nicht wie die höheren Protokolle die Sicherheit mit TLS hinzufügen oder der TLS-Handshake zu initiieren ist (<https://tools.ietf.org/html/rfc5246#section-1>).

### TLS Handshake

Der TLS Handshake ist der Start zum Schlüsselaustausch und ab dann wird verschlüsselt zwischen den beiden Parteien kommuniziert.

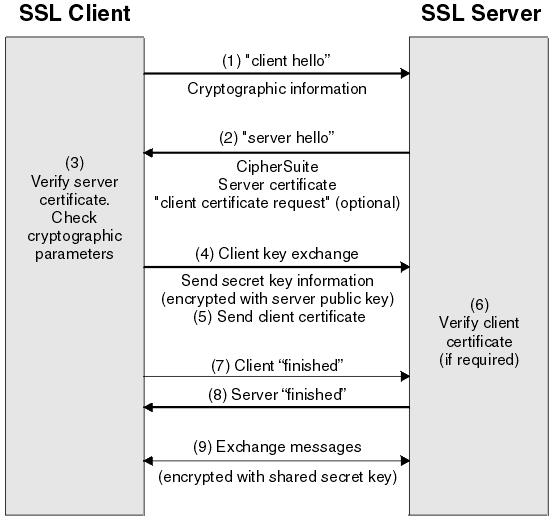


Abbildung 1: TLS Hanshake.jpg [[1]](#_CTVL001580143ecd2134301baf4e216f79fc6b3)

### TLS Record

## Windows Certificate Trust Store

Der Windows Zertifikatsspeicher kann über die Zertifkatsverwaltung aufgerufen werden und dient zur Auflistung der Vertrauenswürdigen und nicht vertrauenswürdigen Zertifikate. Wie aber Tom Aafloen in einem Blogbeitrag von Microsoft Security Solutions [[4]](#_CTVL0017c99aedcbcf44a1393d3676581811584) beschreibt wird in der Zertifikatsverwaltung von Windows nur ein Teil der Zertifikate angezeigt, denen der User auch vertraut. Anhand eines Beispiels, bei der er eine Website aufruft die aktuelle Jobs in Hong Kong zeigt, verdeutlicht er, dass falls eine Website von einer CA signiert wurde, die nicht in der lokalen Zertifikatsverwaltung enthalten ist, wird eine Verbindung zu Windows Update hergestellt, um zu prüfen ob es dort aufgeführt ist. Ist dies der Fall wird es dem lokalen Speicher hinzugefügt. Aafloen betont am Ende seines Beitrages, er wisse nicht, ob das „verstecken“ der vertrauenswürdigen CA’s den Benutzern nicht einen falschen Eindruck über die Sicherheit oder Kontrolle gibt.

Es sollte zumindest jedem absolut klar sein, dass das, was in der Oberfläche der Zertifikatsverwaltung nur eine Teilmenge aller Zertifikate ist.

Ist es dann überhaupt möglich die Zertifikate über die Zertifikatsverwaltung feingranularer zu definieren, wenn der Abgleich mit einer Onlinedatenbank die Gültigkeit von Zertifikaten verifiziert?

# Hauptteil

Im ersten Abschnitt dieses Kapitels wird der Stand der Forschung erläutert, welche Möglichkeiten in Frage kommen können, um das Vertrauen in die Zertifizierungsstellen nicht völlig in deren Hände zu legen.

## Stand der Forschung

Wie bereits oben kurz erläutert ist von Google’s Certificate Transperancy (CT) eine gute Alternative die Sicherheit basierend auf HTTPS und dem SSL/TLS Protokoll zu erhöhen. Der grundlegende Ansatz ist, dass jedes Ausgestellte Zertifikate an einen öffentlich zugänglichen Logserver gesendet wird, der die Zertifikate in einer Merkle-Baum-Datenstruktur abspeichert und diese somit fest verwurzelt sind. Ist nun ein Angreifer in Besitz eines gefälschten Zertifikates einer Domain gekommen das schon ein existierendes Zertifikat in den Logserver hat, kann daraufhin über den Wurzelknoten bestimmt werden ob dies das gleiche Zertifikat ist. Dennoch gibt es einige Aspekte zu beachten. CT funktioniert nur wenn auch auf der Website das Flag für den expect CT Header gesetzt ist und es ist nicht in allen Browsern implementiert (Mozilla beschreibt Performanceprobleme beim Einsatz von CT). [[2]](#_CTVL00189262c0c9793494fb5aa41a90a440cea)

Perl, Fahl et al. 2014 [[6]](#_CTVL0013aaf6c163c024423ac9d58b71c58140f) analysierten auf einen internetweiten Datensatz von 48 Millionen HTTPS-Zertifikaten gestützt und diese mit Trust Stores aller großen Browser- und Betriebssystemhersteller verglichen. Dabei konnten Sie 140 CA-Zertifikate identifizieren, die in zwölf Trust Stores aller wichtigen Plattformen enthalten sind und niemals zum Signieren von Zertifikaten verwendet werden, die in HTTPS eingesetzt werden. Basierend auf diesen Erkenntnissen wird vorgeschlagen diese CA-Zertifikate zu entfernen oder einzuschränken.

**Hier noch genauer darauf eingehen was aus diesem paper die conclusion ist und in wiefern ich mich in der thesis davon abgrenze. Echtzeicht, heimnetzwerk,**

In dieser Arbeit soll es aber um die feingranularere Einstellung eines einzelnen PC’s und den Endbenutzer gehen, bei denen sich in jedem Fall noch weitere Einschränkungen treffen lassen können, sollten als in einem großen Universitätsnetzwerk.

## Methodik/Vorgehensweise

Die Datenerhebung findet mittels eines Fritzbox Router im eigenen Heimnetzwerk statt, welches in der Abbildung 2 detailliert dargestellt wird. Dieser ist in der Lage den Netzwerkverkehr mitzuschneiden. Der Netzwerkmitschnitt wird einmal über eine Woche lang mitgeschnitten und für die Evaluation dann nochmalig eine Woche. Da es sich in dieser Thesis auf ein Hausnetzwerk beschränkt reicht diese Menge an Netzwerkverkehr.

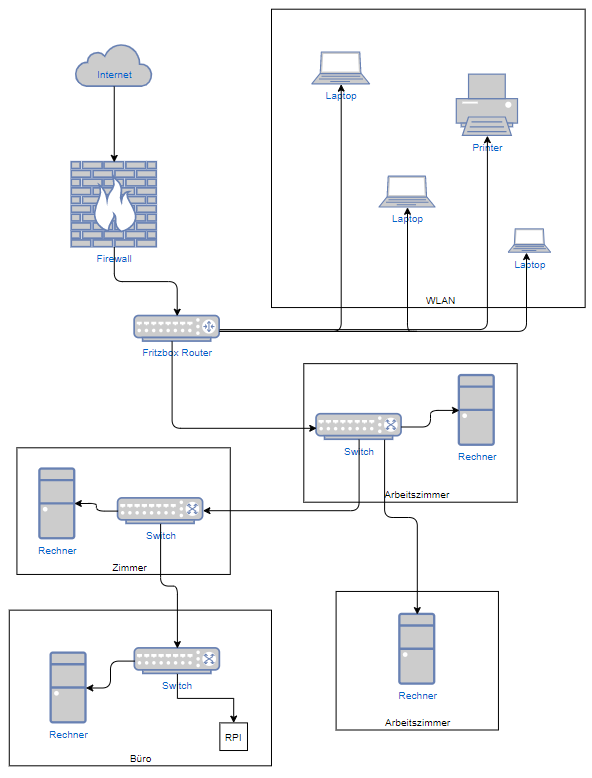


Abbildung 2: Aufbau des Heimnetzwerks

Das Heimnetzwerk ist in diesem Fall ein relativ großes. Alle IP Adressen werden anonymisiert. Ebenso

**Zwei Ansätze der Implementierung**

* C/C++ => Benötigte Bibliotheken: Pcapplusplus, Matplotplusplus, openssl, pthreads. Versuch das Programm in der Programiersprache C/C++ zu schreiben. Das Reassemblen der TCP Stream ist hier ein bisschen schwierig. Für die Implementierung werden die Bibliotheken „pcap“ und „openssl“ verwendet. Für erstere gibt es einen Wrapper für C++, das PcapPlusPlus Projekt [[5]](#_CTVL0010e6234adcf88411baf750b7d038802a0). Somit kann Objektorientiert und modular entwickelt werden, was die Erweiterbarkeit und Wartbarkeit steigern.
* Python => Benötigte Abhängigkeiten: dpkt, netifaces, treelib, cryptography, bs4, requests, coloredlogs, pcapy, matplotlib, numpy. In Python existiert bereits von PeterMosman ein tls protocol analyzer, den ich auf meine bedürfnisse angepasst habe und daraufhin das Extrahieren und Parsen der Zertifikatsketten in die jeweiligen Datenstrukturen.

Entscheidung für die Programmiersprache Python, da es sich mit dieser leichter umsetzten lässt und die Erweiterbarkeit und Lesbarkeit des Codes besser ist.

Struktur für eine Aufstellung einer Zertifizierungsstellenkette, um einen graphischen Eindruck von dieser zu erhalten. Des Weiteren soll eine graphische Aufbereitung aller Zertifikatsketten inklusiv den intermediaten Zertifikaten als „Baum“ dargestellt werden.

Um die Überflüssigen Zertifizierungsstellen in den Systemen zu entfernen (verschieben), wird ein Skript erstellt, das diese feingranulare Einstellung automatisiert übernommen werden kann.

Zur Evaluierung soll am Ende überprüft werden, ob die feingranulare Einstellung der Zertifizierungsstellen ohne Einschränkungen des Endbenutzers stattfindet, d. h. es wird erneut über einen Zeitraum von 2 Wochen mittels dem Fritzbox Router Netzwerkverkehr mitgeschnitten.

## Implementierung

### Anforderungsanalyse

* Die Zertifikate der Root Zertifizierungsstellen sollen in das Programm eingelesen werden.
* Das Programm soll über Argumenten Parameter entweder eine Pcap-datei erhalten oder eine Netzwerkadresse, an der das Programm dann in Echtzeit den Netzwerkverkehr mitschneiden kann.
* In beiden Fällen von zwei (Datei/Netzwerkadresse) sollen dann die TLS-Handshake Nachrichten extrahiert werden (-> inspiriert/angelehnt an Peter Moosmann GitHub)
* Im Verlauf des Programms werden dann die Zertifikatsnachrichten expliziert inspiziert und das jeweilige Root Zertifikat gesucht.

### Aufbau und Struktur

Des Weiteren muss zu Beginn ein Prototyp einer C/C++ Anwendung implementiert werden, der bestimmte Teilbereiche aus dem Netzwerkverkehr extrahiert und in eine Form bringt, um statistische Auswertungen möglich zu machen. Die Implementierung soll alle TLS Handshake Pakete finden und daraus die Zertifikate extrahieren. Da das TLS Protokoll auf Basis der TCP Protokoll übertragen wird, muss ein Reassemblen der Pakete sichergestellt werden. Die Struktur der Daten wird wie folgt aufgebaut:

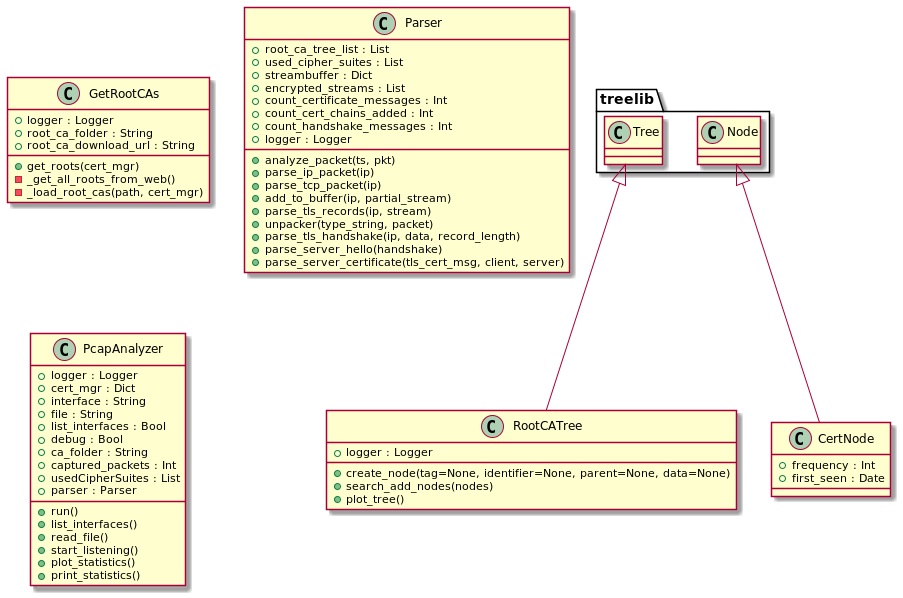


Abbildung 3: Klassendiagram PcapAnalysis

### Probleme bei der Implementierung?

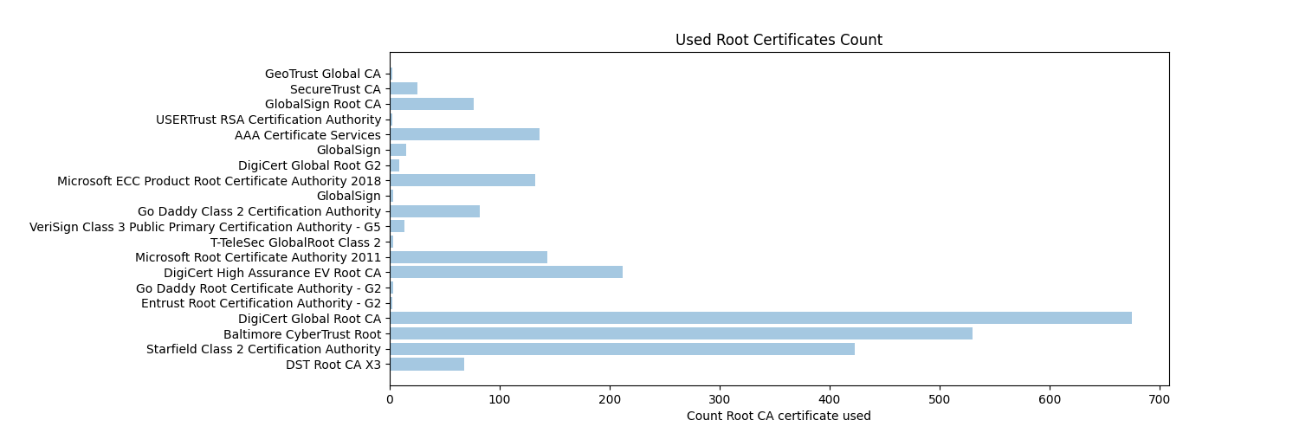
* Reassemblen des TCP Streams in C++ nicht so einfach, die reassembly Methoden des PcapPlusPlus Wrappers (bestehend aus 3 Callbacks) speichern die streams nicht als Ganzes sondern als Pakete in einem Array, sodass im Nachhinein die Pakete konkateniert werden müssen => In Python existiert eine Methode tls\_multi\_factory welche einen stream als Parameter bekommt und daraus die einzelnen TLS Records erstellet (automatisch 😊)

## Ergebnisse

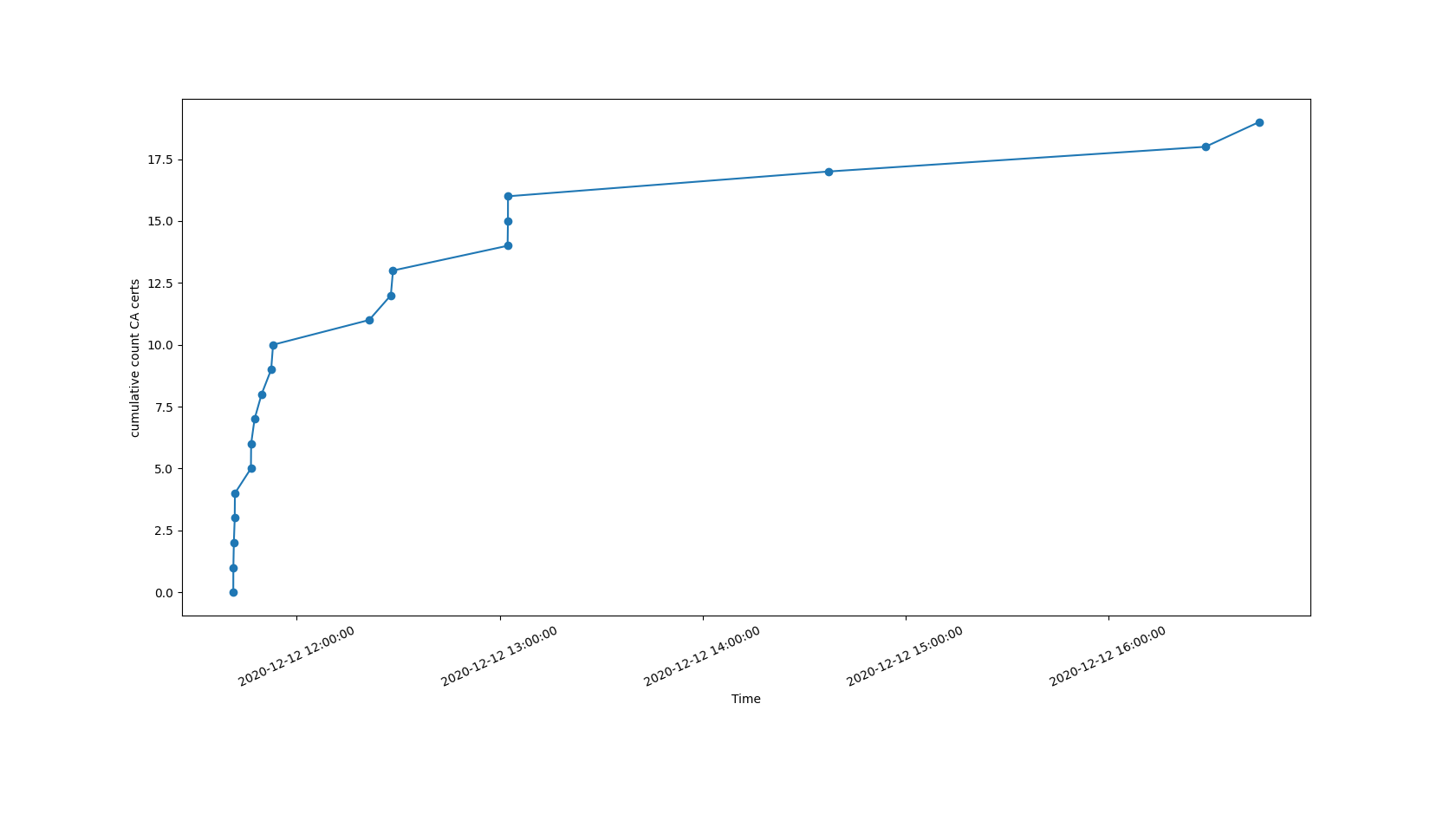
Bei der Analyse von Pcap Dateien kann es vorkommen das manche Webseiten noch mit einem älteren Root CA Zertifikat signiert wurden und manche schon mit einer neuen Version. Als Beispiel hier USERTrust RSA Certification Authority in der fritzbox.pcap Datei.

Für die statistische Auswertungen sollen folgende Fragen beantwortet werden:

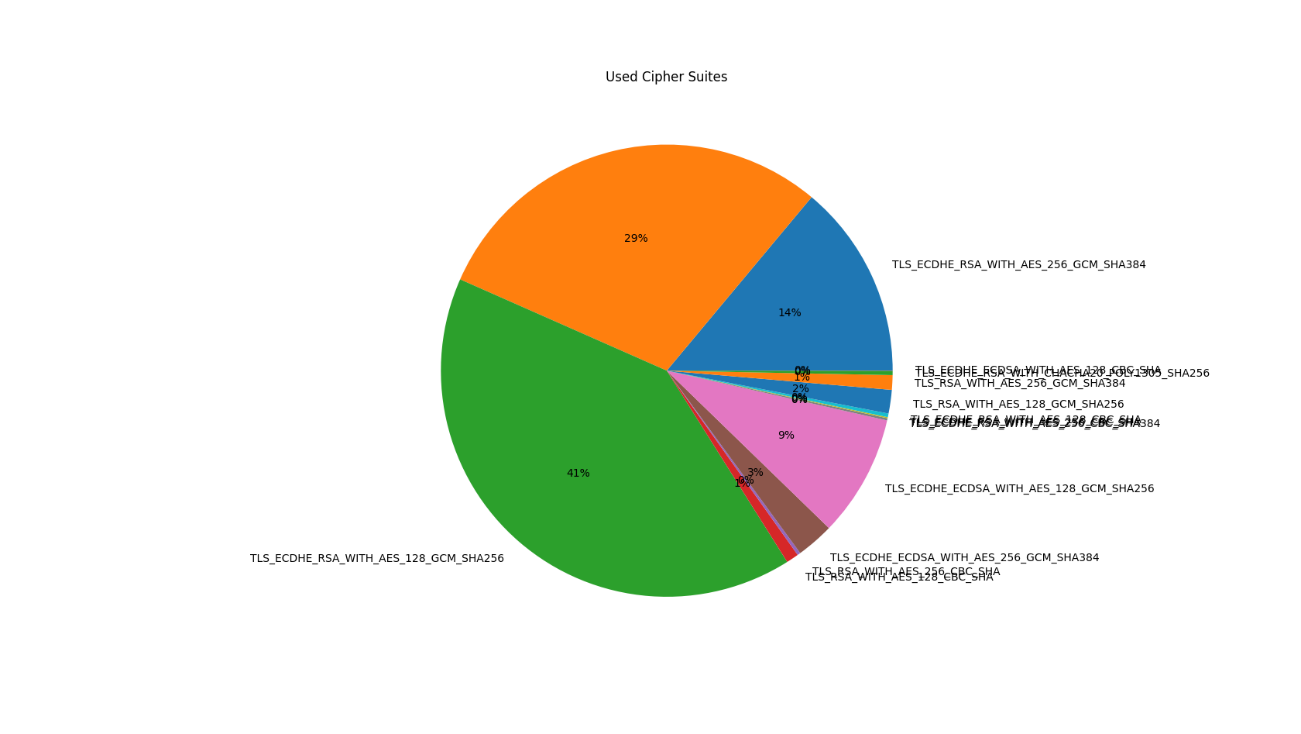
Werden alle hinterlegten Zertifizierungsstellen verwendet? Wie viele davon werden verwendet?



Auf welche Zeit konvergiert die Anzahl der CA’s?



Welche CipherSuites verwenden die meisten Server?



Welche angesurften Seiten waren dabei die kein Zertifikat enthalten haben? Und wie viele?

Gibt es Springer in den Domains (Wechsel der Server)?

Was war die Frage der Thesis?

Wie wurde sie beantwortet? Was sind die konkreten Erkenntnisse aus den verschiedenen Statistiken?

Welche neuen Fragen gibt es?

# Fazit

## Zusammenfassung

## Reflexion

## Ausblick

Literaturverzeichnis

[1] 2014. *An overview of the SSL or TLS handshake*. https://​www.ibm.com​/​support/​knowledgecenter/​en/​SSFKSJ\_8.0.0/​com.ibm.mq.sec.doc/​q009930\_.htm. Accessed 1 March 2021.

[2] 2020. *Certificate Transparency*. https://​www.certificate-transparency.org​/​. Accessed 7 July 2020.

[3] Kloep, P. 2020. *PKI und CA in Windows-Netzwerken*. *Das umfassende Handbuch*. Rheinwerk Computing.

[4] Microsoft Security Solutions. 2013. *Which Root CAs do you really trust?* https://​mssec.wordpress.com​/​2013/​01/​31/​which-root-cas-do-you-really-trust/​. Accessed 29 December 2020.

[5] 2020. *PcapPlusPlus - a multiplatform C++ library for capturing, parsing and crafting of network packets*. https://​pcapplusplus.github.io​/​. Accessed 7 July 2020.

[6] Perl, H., Fahl, S., and Smith, M. 2014. You Won’t Be Needing These Any More: On Removing Unused Certificates from Trust Stores. In *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, Berlin, Heidelberg, 307–315.

[7] 2021. *Referenzmodelle und Protokolle der Netzwerktechnologie - InfoTip Kompendium*. https://​kompendium.infotip.de​/​netzwerktechnologie2-referenzmodelle-und-protokolle.html. Accessed 1 March 2021.

**Abbildungsverzeichnis**

[Abbildung 1: TLS Hanshake.jpg [1] - 13 -](#_Toc65828744)

[Abbildung 2: Aufbau des Heimnetzwerks - 17 -](#_Toc65828745)

[Abbildung 3: Klassendiagram PcapAnalysis - 19 -](#_Toc65828746)

**Listingverzeichnis**

**Anhang**