INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Flow-Record-Fingerprinting

Host-, Dienst- und Software-Klassifikation basierend auf Flow-Records

Christian Andreas Werner Walonka

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Flow-Record-Fingerprinting

Host-, Dienst- und Software-Klassifikation basierend auf Flow-Records

Christian Andreas Werner Walonka

Aufgabensteller:	Prof. Dr. Helmut Reiser
Betreuer:	Daniela Pöhn Michael Grabatin
Abgabetermin:	22.11.2016

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 22.11.2016

. . . . . .

(Christian Andreas Werner Walonka)

Abstract

Rechnernetze sind unbestritten ein wesentlicher Bestandteil des täglichen Lebens geworden. So werden diese in Hochschulnetzen, Unternehmen sowie im privaten Umfeld eingesetzt. Das Wissen um die sich in einem Rechnernetz befindliche Infrastruktur zählt hierbei zu einer der wichtigsten Informationen. Oftmals ist dieses Wissen jedoch begrenzt und über mehrere Wissensträger, beispielsweise verschiedene Administratoren oder Abteilungen verstreut, oder gar nicht vorhanden.

Interesse an diesem Wissen über die eingesetzte Infrastruktur haben verschiedene Akteure. Im Rahmen dieser Arbeit zählen zu diesen Akteuren Hochschulrechenzentren, Unternehmen sowie Strafverfolgungsbörden und Nachrichtendienste. Insbesondere sind die eingesetzten Betriebssysteme, die dort betriebene Software sowie die bereitgestellten Dienste von großem Interesse. Ebenfalls besteht Interesse an den Informationen darüber, welche Versionen (Patchstand) hinsichtlich Betriebssystem und Software im Einsatz sind.

Häufig ist die Erstellung einer vollständigen IT-Asset-Datenbank, in der diese Informationen enthalten sind, mit Hilfe einer aktiven Netzanalyse aufgrund der Beeinträchtigungen im Netz nicht möglich und die im Vergleich dazu anwendbare passive Untersuchung des vollständigen Netzverkehrs wird meist aufgrund der hohen Kosten nicht durchgeführt. Im Rahmen dieser Arbeit wird die Analyse von Rechnernetzen mit Hilfe der leichtgewichtigen und damit kostengünstigen Flow-Records untersucht, um Rückschlüsse auf genutzte Betriebssysteme und eingesetzte Software zu erhalten. Hierfür wird ein in den produktiven Einsatz übertragbares Konzept geliefert und im Anschluss mit Hilfe eines Proof of Concept evaluiert. Auf Basis der erfolgreichen Evaluation werden die erkannten Einschränkungen angesprochen und hierauf aufbauend ein Ausblick für weitere Anpassungen gegeben.

Inhaltsverzeichnis

1. Einleitung

Das Wissen über in einem Rechnernetz eingesetzte Betriebssysteme sowie Software ist für Netzverantwortliche von unentbehrlichem Vorteil. Unter der Voraussetzung einer vollständigen Dokumentation sämtlicher sich im Netz befindlichen Systeme lassen sich dieses Wissen und weitere Informationen wie beispielsweise Rechnername und betriebene Dienste aus einer (IT-)Asset-Datenbank entnehmen [Wö]. Eine stets aktualisierte (IT-)Asset-Datenbank ermöglicht unter anderem veraltete Software gezielt und schnell zu erkennen, Back-ups zu konfigurieren oder den Netzausbau beziehungsweise den Netzumbau gezielt zu planen [GvE16].

Oftmals ist eine allen Verantwortlichen vorliegende Asset-Datenbank jedoch mangelhaft oder überhaupt nicht vorhanden, da häufig aufgrund von Zeitmangel nicht überall eine Dokumentation bei der Bereitstellung, der Aktualisierung oder der Abschaltung entsprechender Dienste erfolgt. Zudem können Teilnehmer im Netz (User) – abhängig von den jeweiligen Systemrichtlinien – selbst Software installieren, unternehmensfremde Geräte mit dem Netz verbinden oder Anwendungen für Dritte bereitstellen (Schatten-IT). Um dennoch Sicherheit, Verfügbarkeit und Qualität des Netzes sowie der betriebenen Dienste jederzeit und ohne Risiko gewährleisten zu können, benötigen die Netzverantwortlichen alternative und aussagekräftige Datenquellen [GvE05, GvE16].

Die Anforderungen nach korrektem und aktuellem Datenmaterial ergeben sich überall dort, wo Netze betrieben werden. Im Rahmen dieser Arbeit wird detaillierter auf die Netze von Hochschulrechenzentren, allgemeiner auf Firmennetze sowie Strafverfolgungsbehörden und Nachrichtendienste eingegangen.

Für das Szenario der Hochschulrechenzentren wird exemplarisch das Leibniz-Rechenzentrum (LRZ) betrachtet. Das LRZ betreibt dabei das Münchner Wissenschaftsnetz (MWN), welches die Münchner Universitäten, deren Institute sowie weitere Einrichtungen wie beispielsweise das Studentenwerk München miteinander verbindet. In diesem Netz mit über 100.000 Teilnehmern und 300.000 unterschiedlichen Geräten ist das LRZ für Bereitstellung und Betrieb des Netzes sowie verschiedener bereitgestellter Dienste verantwortlich [LR15, LR16]. Im Gegensatz zu einem Firmennetz, in welchem die Verantwortlichen oftmals das vollständige Netz betreuen, endet der Einflussbereich des LRZ gegebenenfalls schon mit der Zuteilung von IP-Adressen und Hostnamen. Dies ist in der weitreichenden Freiheit der Lehrstühle, Institute und angebundenen Organisationen begründet, die unter anderem eine Vielzahl von Netzdiensten und Systemen ohne Wissen oder Beteiligung des zugehörigen Rechenzentrums betreiben. Nach außen ist das LRZ jedoch Ansprechpartner für Sicherheitsvorfälle oder Probleme aller Art [GvE16].

Seit 1994 das Konzept der Betriebssystemidentifizierung auf Basis von Netzverkehr vorgestellt wurde, kristallisierten sich zwei Hauptrichtungen heraus. Dabei handelt es sich um die aktive und die passive Analyse des Netzes [Jv14], die nachfolgend kurz vorgestellt werden.

Aktive Analysetechniken scannen, von einem Ausgangspunkt, wie beispielsweise einem Rechner im Netz, das für sie erreichbare Teilnetz. Hierbei kann an jeder erreichbaren IP-Adresse geprüft werden, ob ein Teilnehmer antwortet und – falls ja – festgestellt werden, welche Ports unter dieser IP-Adresse nach außen geöffnet sind. Dies ist unter anderem mit

einem Tool wie Nmap möglich, das verschiedene TCP-, UDP- und ICMP-Anfragen versendet. Zentrales Problem an diesem Ansatz ist, dass die Auslastung des Netzes hinsichtlich dem Datendurchsatz durch die aktive Detektion erhöht wird, so dass die Netzqualität (Verfügbarkeit, Geschwindigkeit, Verzögerung) hierdurch negativ beeinflusst werden kann. Des Weiteren kann dieser Ansatz die gescannten Systeme in ihrem Betrieb durch störende Log-Einträge behindern oder Router durch eine hohe Anzahl an SYN-Anfragen überlasten [GvE16]. Auch ist ein derartiger Eingriff ohne Zustimmung der Betroffenen rechtlich fragwürdig [Jv14] und wird von einigen Systemen als Angriff erkannt, geblockt oder nicht beantwortet.

Die passive Detektion hingegen basiert auf der reinen Betrachtung des anfallenden Netzverkehrs, wobei im allgemeinen Fall jedes Netzpaket durch eine Analysesoftware überprüft wird. Vorteil dieser Methode ist, dass Nutzer hierbei weder gestört noch beeinflusst (Verzögerungen, zusätzliche Logdateien, Rechenlast des Zielsystems) werden. Als Einschränkung für die passive Detektion ist an dieser Stelle anzumerken, dass lediglich Inhalte unverschlüsselter Verbindungen genau untersucht sowie nur aktiv am Netz teilnehmende Geräte erkannt werden können [GvE16, Jv14].

Die Methode der passiven Detektion lässt sich in verschiedene weitere Verfahren unterteilen. Die maximale Ausprägung nutzt den vollständigen Netzverkehr, während weniger ausgeprägte Verfahren lediglich die anfallenden Metadaten/Flow-Records beziehungsweise eine Mischform nutzen. Jede dieser Formen geht mit spezifischen Vor- und Nachteilen einher.

Eine vollständige Ausleitung und Analyse des anfallenden Datenverkehrs ist zwar am genauesten, birgt jedoch erhebliche Kosten in sich. Diese entstehen durch die oft kostspielige Ausleitung des gesamten Datenverkehrs an einem Mirror-Port und deren Weiterleitung an die entsprechende Analyseplattform sowie der hierfür benötigten Performance. Des Weiteren ist diese Methode aufgrund von Datenschutzbestimmungen sowie gesetzlicher Regelungen in der Breite ohne explizite Zustimmung aller Nutzer oftmals nicht durchführbar [GvE16].

Bei einer Mischform von Flow-Record-Analyse und vollständiger Analyse des Datenverkehrs werden beispielsweise die ersten 100 Bytes eines jeden Netzpakets für die Analyse herangezogen. Alternativ ist es auch möglich, die entstandenen Metadaten sowie die Domain Name Service (DNS)-Anfragen zu nutzen.

Aus den anfallenden Metadaten wie Quell- und Ziel-IP, Übertragungsgröße, genutztem Protokoll und auch verwendeten Ports lassen sich in Verbindung mit den DNS-Anfragen, welche beispielsweise den Hostnamen de.archive.ubuntu.com beinhalten, ableiten, dass das anfragende System mit einer gewissen Wahrscheinlichkeit Ubuntu nutzt. Mit einem derartigem Verfahren ist es zudem möglich, Wahrscheinlichkeiten für genutzte Software abzuleiten.

Aus den vorhergehend genannten Analysemöglichkeiten ergibt sich die auf Flow-Record-Analyse basierende nachfolgende Aufgabenstellung, deren Teilziele sowie die Struktur der Arbeit.

1.1. Aufgabenstellung und Ziele

Im Rahmen dieser Arbeit soll ein neuartiger Ansatz zur passiven Netzanalyse vorgestellt und analysiert werden.

Hierfür wird der von Felix von Eye und Michael Grabatin im Rahmen der 23. DFN-Konferenz „Sicherheit in vernetzten Systemen“ publizierte Ansatz aus „Passive Detektion von Betriebssystem und installierter Software mittels Flow-Records“ genutzt und weiter ausgebaut. Dieser Ansatz basiert auf Verwendung von Flow-Records als Datenbasis zur Erkennung von Übertragungsinhalten [GvE16].

Nachfolgend wird untersucht, ob und wie weit auf diesem Ansatz aufbauend, das Betriebssystem, betriebene Dienste sowie die genutzte Software mit Hilfe einer Flow-Record-Analyse erkannt werden können.

Die Auswertung der durch diese Analyse gewonnenen Daten ergibt eine Aussage, ob die Anwendung des hierfür neu erstellten Flow-Record-Fingerprinting-Tools (FRF-Tools) eine verwertbare Möglichkeit ergibt, um sowohl Betriebssystem als auch betriebene Dienste und eingesetzte Software zu erkennen. Der Inhalt dieser Arbeit kann damit auch wie folgt definiert werden: Wie weit und wie genau ist mit Hilfe der Analyse von Flow-Records über ein noch zu entwickelndes prototypisches FRF-Tool eine Detektion des Betriebssystems sowie betriebener Dienste und installierter Software möglich?

Da zunächst, um bewertbare Ergebnisse über das neu zu konzeptionierende FRF-Tools zu erhalten, die genauen Anforderungen definiert werden müssen, gliedert sich der strukturelle Rahmen dieser Arbeit in verschiedene Teilziele.

Ausarbeitung der Anforderungen
Für die Konzeptionierung, Erstellung und Evaluierung des im Rahmen dieser Arbeit entwickelten FRF-Tools ist es notwendig, konkrete Anforderungen herzuleiten. Dies geschieht anhand von drei ausgewählten Szenarien (Szenario 1: Hochschulrechenzentrum, Szenario 2: Unternehmen, Szenario 3: Strafverfolgungsbehörden und Nachrichtendienste).

Abgrenzung zu themenverwandten Arbeiten
Um zu überprüfen, ob und inwieweit bisher veröffentlichte Verfahren die im Rahmen dieser Thesis erarbeiteten Anforderungen erfüllen beziehungsweise inwiefern eine Kombination bisher bekannter Verfahren die Anforderungen erfüllt, wird eine Analyse ausgewählter bisheriger Verfahren durchgeführt. Des Weiteren wird ein Abgleich dieser mit den ausgewählten Gesamtanforderungen durchgeführt.

Konzeption eines geeigneten Detektionsverfahrens
Die Basis des FRF-Tools stellt die Ausarbeitung eines Konzeptes dar, in welchem die technischen Rahmenbedingungen und die strukturelle Herangehensweise erläutert werden. Neben diesen technischen Anforderungen an ein Werkzeug zur Wiedererkennung von Übertragungsinhalten erfordern insbesondere die Auswahlkriterien der Datenbasis in Form von Flow Records und Vergleichswerten sowie die später mittels des Sample-Generators erstellten und gesammelten Daten eine Darstellung und Begründung.

Erstellung eines Testnetzes zur Generierung von Daten
Um ein Training des FRF-Tools sowie dessen Evaluation zu ermöglichen, wird ein Testnetz erstellt. Die einzelnen virtualisierten Endgeräte werden hierbei mit verschiedenen Betriebssystemen installiert. Neben den verschiedenen Betriebssystemen wird eine Auswahl an Anwendungen und Diensten bereitgestellt. In diesem auf virtualisierter Hardware aufgebauten Netz übernimmt der Edge-Router die Aufgabe der Erfassung der Flow-Records und stellt gleichzeitig den Knotenpunkt nach außen dar.

Erstellung einer geeigneten Datenbankstruktur
Für die Speicherung und Verwertung der gesammelten Flow-Records wie auch der Trainingsdaten ist ein geeignetes Datenbankschema zu erstellen. Innerhalb dieses Schemas sollen ebenfalls die Ergebnisse der Analyse in Form einer exemplarischen Asset-Datenbank durch das FRF-Tool gesichert werden können.

Entwicklung und Training des FRF-Tools
Zur Interpretation der im Testnetz generierten Flow-Records wird ein FRF-Tool entwickelt. Für eine Erkennung beziehungsweise Wiedererkennung ist es notwendig, dieses FRF-Tool mit Hilfe von bekannten Daten (zum Beispiel um Informationen über den betroffenen Host erweiterte Flow-Records) zu trainieren. Hierfür erfolgen durch einen Sample-Generator gezielte Übertragungen mit bekannten Inhalten. Nachdem das entsprechende Tool mit diesen Daten trainiert wurde, wird das Wiedererkennen der trainierten Datensätze anhand der in den anfallenden Flow-Records enthaltenen Metadaten wie Größe, genutzte Ports und IPs untersucht.

Analyse der erfassten Daten – Rückschlüsse auf Betriebssystem, betriebene Dienste und installierte Software
Für eine Erkennung der genutzten Betriebssysteme wie auch betriebener Dienste und eingesetzter Software ist es notwendig, im Netz übertragene Daten zu erkennen beziehungsweise rekonstruieren zu können. Bei diesen Daten kann es sich beispielsweise um Betriebssystem- oder Softwareupdates, die Installation systemspezifischer Software oder systemspezifische Anfragen an einen Updateserver beziehungsweise laufenden Dienst handeln. Auch können softwarespezifische Antworten auf Anfragen, zum Beispiel den Aufruf einer Webseite, in den Daten vorhanden sein und genutzt werden. Hierfür soll das FRF-Tool Aussagen über den vermutlichen Inhalt beziehungsweise angesprochenen Dienst einer Übertragung treffen.

Evaluation der Eignung statistischer Analyseverfahren
Um ein geeignetes Analyseverfahren zu ermitteln, erfolgt eine Evaluation verschiedener Algorithmen und Verfahren aus dem Bereich der Künstlichen Intelligenz und Stochastik. Dafür werden Entscheidungsbäume, Neuronale Netze oder die Bayes-Analyse untersucht. Hieraus ergibt sich ein geeignetes Set aus Verfahren, welches die Analysebasis des FRF-Tools bildet.

Ausblick und Erweiterung des Konzeptes
Reicht die Qualität der Prognosen aus dem ursprünglich konstruierten Ansatz nicht aus, werden – ohne das ursprüngliche Konzept der Leichtgewichtigkeit aus den Augen zu verlieren – verschiedene Empfehlungen für eine Erweiterung des Konzepts als Ausblick geliefert.

1.2. Struktur der Arbeit

Aufgaben und besondere Teilziele sind bereits in Kapitel 1.1 genannt, so dass nachfolgend lediglich der Aufbau dieser Arbeit erläutert wird.

In Kapitel 2 werden Grundlagen, die für diese Arbeit benötigt werden, definiert. Es handelt sich hierbei insbesondere um Flow-Records, Fingerprinting, Erläuterungen zu IT-Asset Management und den genutzten Sample-Generator.

Um die Anforderungen an diese Arbeit zu definieren, beginnt das 3. Kapitel mit Fallbeispielen. Diese bedienen sich der drei vorhergehend genannten Szenarien, nämlich Hochschulrechenzentren, Unternehmen sowie Strafverfolgungsbehörden und Nachrichtendiensten. Der sich hieraus ergebende unterschiedliche Bedarf an ein FRF-Tool wird über tabellarische Ansichten ausgewertet und im Anschluss priorisiert. Durch die Diskussion der Individualanforderungen ergeben sich die konkreten Gesamtanforderungen an das Analysetool dieser Arbeit.

Zur Abgrenzung von themenverwandten Arbeiten und um den Stand der Technik darzustellen, gibt Kapitel 4 kurze Abrisse der jeweils wichtigsten themenverwandten Arbeiten. Diese Arbeiten werden mit den konkreten Gesamtanforderungen an das hier beschriebene Analysetool abgeglichen. Anschließend wird ein Überblick über die Eignung der verschiedenen Verfahrensklassen gegeben.

Das Konzept dieser Arbeit wird in Kapitel 5 in drei Hauptschritten beschrieben. Hierbei wird zunächst auf die Datenerfassung, -Übertragung und -Speicherung eingegangen. Im Anschluss wird die Klassifizierung von Hosts, betriebenen Diensten sowie genutzter Software erläutert. Abschließend folgt die Beschreibung der Datenauswertung und -Speicherung.

Kapitel 6 enthält Details zur Konfiguration der eingesetzten Software sowie Implementierung des FRF-Tools. Hierfür wird auf das Labornetz sowie die Erzeugung und Verarbeitung von Trainingsdaten eingegangen. Des Weiteren werden die Datenerfassung, -Übertragung und Speicherung im Versuchsaufbau beschrieben. Im Anschluss werden die Datenklassifizierung in den Bereichen Host, Dienst und Software sowie die Datenauswertung und Sicherung der Ergebnisse beschrieben.

Die in den Tests festgestellten Ergebnisse werden in Kapitel 7 diskutiert. Des Weiteren erfolgt in diesem Kapitel ebenfalls ein Abgleich mit den in Kapitel 3 ermittelten Gesamtanforderungen.

Zum Abschluss werden in Kapitel 8 die Ergebnisse dieser Arbeit kurz zusammengefasst. Um eine potentielle Verbesserung der Erkennungsrate zu ermöglichen, erfolgt zudem ein Ausblick auf mögliche Erweiterungen des in Kapitel 5 vorgestellten Konzepts.

2. Grundlagen

In diesem Kapitel werden grundlegende Begrifflichkeiten erläutert, die für die weitere Arbeit relevant sind.

2.1. Flow-Records

Unter Flow-Records versteht man eine Datenstruktur, auf der aufbauend Protokolle wie Netflows[1], sFlows[2], sowie weitere Implementierungen umgesetzt werden. Abhängig von der Implementierung können auch mehr Daten, als in der ursprünglichen Datenstruktur vorgesehen, enthalten sein. Im Rahmen dieser Arbeit wird die grundlegende Datenstruktur der Flow-Records genutzt.

Bei der Erfassung von Flow-Records handelt es sich um eine passive Erkennungsmethode, durch die der Datenverkehr nicht beeinflusst wird. Die Aufzeichnung (und Speicherung) der Datenströme wird durch einen Flow-Sammler durchgeführt, der vom Router oder anderer Netz-Infrastruktur (Flow-Exporter) exportierte UDP-Datagramme erhält. Die so gespeicherten Flow-Records werden im Anschluss auf einem Analyzer ausgewertet (vgl. Abbildung 2.1).

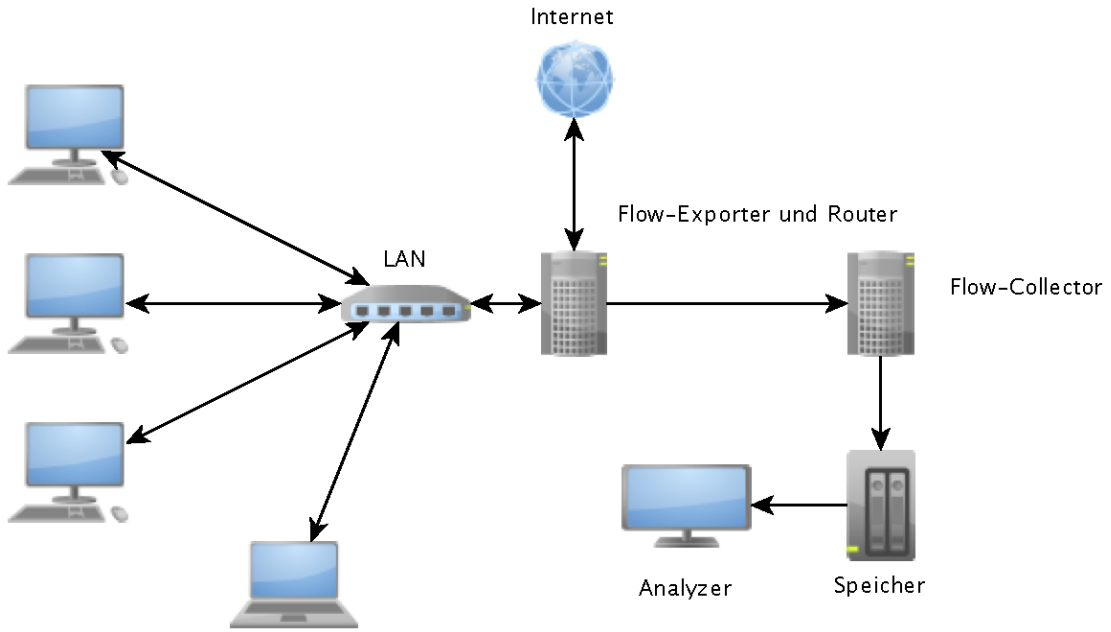


Abbildung 2.1.: Schematische Darstellung der Erfassung und Verarbeitung von Flow-Records

[1] http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html
[2] http://www.sflow.org

Die Erfassung der Flow-Records kann in allen Netzkomponenten, die durchlaufen werden, durchgeführt werden. Wie in Abbildung 2.2 dargestellt, können diese Daten innerhalb der Kollisionsdomäne eines lokalen Netzes (1) durch jeden Teilnehmer gesammelt werden. Des Weiteren können Flow-Records in vielen Switches (2,3) oder auch dem Edge-Router (4) sowie auf dem Transportweg ins Internet (5) erfasst werden.

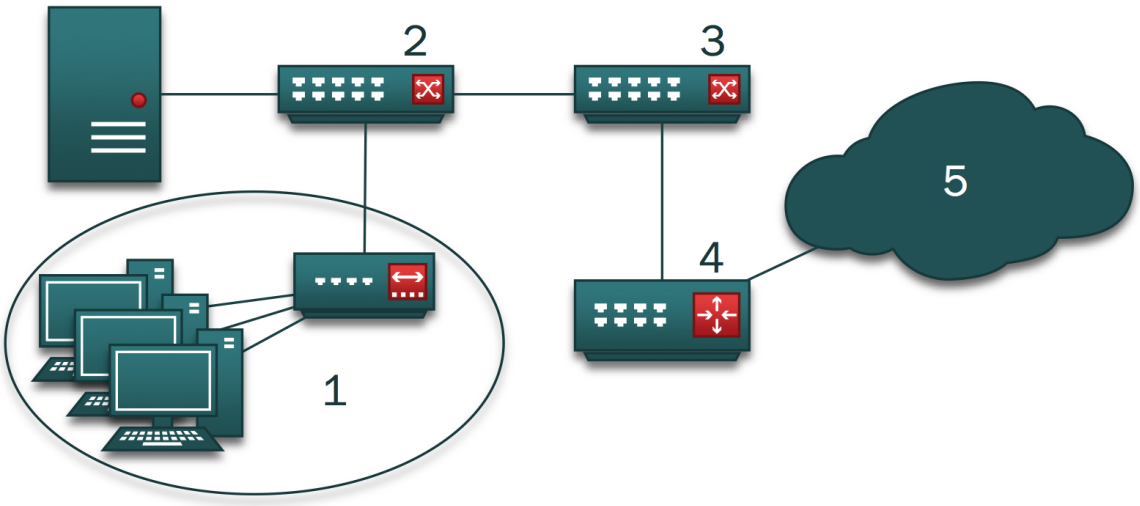


Abbildung 2.2.: Schematische Darstellung eines Netzes [GvE05]

Das vollständige Mitschneiden des Netzverkehrs ist beispielsweise mit Tools wie *tcpdump* oder *Wireshark* möglich. Aus den so aufgezeichneten Daten können anschließend Flow-Records gewonnen werden.

Mit Tools wie softflowd sowie *flow-capture* aus der *flow-tools* Programmsammlung lassen sich auch direkt im laufenden Betrieb Flow-Records sammeln [GvE16, GvE05]. In einem so aufgezeichneten Flow-Record sind unter anderem folgende Daten enthalten:

- Zeitstempel
- Quell- und Ziel-IP-Adressen
- Quell- und Ziel-Port
- TCP-Flags
- Protokoll-Typ

Für diese Arbeit wird der gesamte Verkehr, der eine Netzkomponente durchläuft, über einen definierten Zeitraum aufgezeichnet. Durch eine ausreichend lange Aufzeichnung ist es möglich, nicht nur verkehrsreiche Dienste, sondern auch Dienste mit geringem Traffic entdecken zu können.

2.2. Fingerprinting

Scotland Yard führte im Jahr 1901, noch vor Existenz der ersten Computer, ein Fingerabdruckverfahren ein, um Straftäter in Zukunft eindeutig überführen zu können. Dieses Verfahren unterscheidet sich zwar von der heutigen probabilistischen Definition des Fingerprintings, dient aber aufgrund der weitreichenden Gemeinsamkeiten zu aktuellen Hashingverfahren als Einstieg in die Materie des Fingerprintings. So nahm Scotland Yard von allen überführten Straftätern Fingerabdrücke in ihre Kartei auf. Sofern an einem Tatort Fingerabdrücke gefunden wurden, verglich man diese von Hand mit den in der Kartei hinterlegten Abdrücken, wodurch am 13. September 1902 der erste Verbrecher überführt werden konnte [Cap90, LAM77, Wal14].

Wie im damaligen Verfahren, in dem die Identifikationsmerkmale eines Verdächtigen auf seine Fingerabdrücke zur eindeutigen Identifizierung reduziert werden konnten, wird dieses Grundprinzip auch heute genutzt, um große Datenmengen in kleinere, leichter zu vergleichende Datensätze zu überführen [Wal14].

Als Beispiel für die Nutzung von Fingerprinting lässt sich der Vergleich großer Datenmengen bei Annahme von nahezu unbegrenzter (günstiger) Rechenleistung und begrenzter (teurer) Bandbreite zwischen zwei Systemen aufführen. Möchte man beispielsweise für ein Backup feststellen, ob die Dateien auf dem Quell- und Zielsystem identisch sind, gibt es insbesondere die drei nachfolgenden Möglichkeiten.

Bei der ersten und genauesten Möglichkeit werden Dateien auf Backup-Server und zu sicherndem System bitweise miteinander verglichen und bei Unterscheidung ein neues Backup angestoßen. Auf Grund der teuren Übertragung ist dies jedoch nicht praktikabel.

Daneben besteht die Möglichkeit für den Vergleich, nur wenige zufällig ausgewählte Dateien zu übertragen. Bei dieser Methode entstehen durch die geringere benötigte Bandbreite zwar weniger Kosten, dafür besteht jedoch die Gefahr, dass eine geänderte Datei nicht in der Menge der zufällig ausgewählten Dateien enthalten ist und somit fälschlicherweise kein Backup erstellt wird.

Die letzte Methode ist die des Fingerprintings, bei dem von jeder zu prüfenden Datei durch mathematische Verfahren ein Fingerabdruck genommen und zur Überprüfung an den Backup-Server übermittelt wird. Da in diesem Verfahren eine große Datei in einen kürzeren Fingerabdruck überführt wird, besteht das Risiko, dass mehrere Eingaben zur selben Ausgabe führen, weshalb Fingerprinting zu den probabilistischen Verfahren gezählt wird [BK11, Die08].

Unter dem Begriff des OS-Fingerprinting versteht man die Erkennung von Betriebssystemen durch die Beobachtung diverser Reaktionsweisen sowie Charakteristika der sich im Netz befindlichen Systeme aus der Ferne. Zur Erkennung des Betriebssystems können sowohl aktive als auch passive Methoden Verwendung finden.

Bei passiven Methoden wird der anfallende Datenverkehr zwischen Quell- und Zielsystem, der den Beobachter durchläuft, bewertet und analysiert. So kann eine einfache Websitzung durch die gleichzeitige Analyse durch ein passives OS-Fingerprinting untersucht werden. Hierbei können detaillierte Informationen wie eingesetztes Betriebssystem oder genutzter Browser zu einem Zielsystem ermittelt werden. Im Gegensatz hierzu werden bei aktiven Methoden Daten zum Zielhost übertragen, um die resultierende Antwort zu analysieren.

2.3. IT-Asset-Management (ITAM)

Asset-Management bedeutet die Verwaltung von Aktivposten, Anlagegegenständen und -Gütern. In Verbindung mit der Informationstechnologie (IT) handelt es sich im engeren Sinne um das Management von Hard- und Software. Betrachtet man das IT-Management im weiteren Sinne so kann man erkennen, dass die Informationstechnologie heute nahezu in jedem Teil einer Organisation zu finden ist. Abbildung 2.3 verdeutlicht die Weitläufigkeit des Einflussbereichs des IT-Asset-Managements.

Zu den IT-Assets zählen damit auch alle unternehmenseigene Informationen, Systeme und Hardware, die im Verlauf von Geschäftsaktivitäten genutzt werden. Das Ziel von IT-Asset-Management (ITAM) ist die effiziente Verwaltung all dieser genutzten IT-Assets, um unter den Stichworten Lebenszyklus, Beschaffungsmanagement und Beschaffungserfassung bis hin zur Entsorgung, Entscheidungsgrundlagen für Hard-und Softwarekäufe (einschl. Lizenzen) sowie die Steuerung und Optimierung von Prozessen in einer sich ständig und immer schneller wandelnden Umgebung geben zu können.

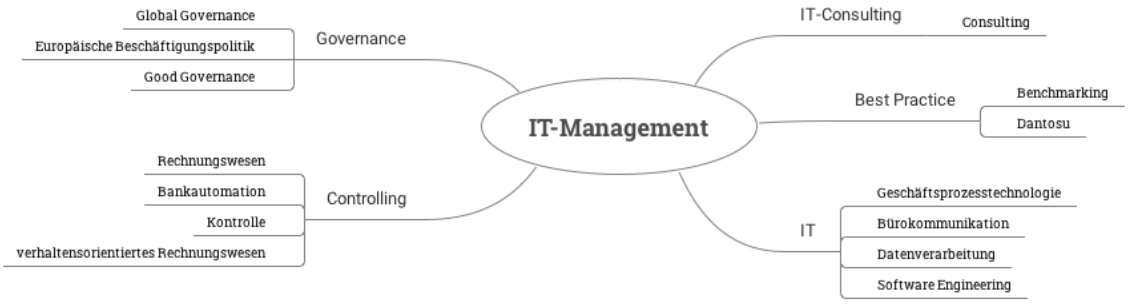


Abbildung 2.3.: Mindmap zur Verdeutlichung der Weitläufigkeit von IT-Asset-Management nach [Gab16]

Zwischenzeitlich gibt es viele Anbieter für ein derartiges Tool. Gemeinsam haben die angebotenen Tools, dass sie

- vorhandene Bestände aufspüren und die Erstellung von entsprechenden Inventarlisten ermöglichen (Bestandsverwaltung)
- Mietbestände, Garantien, Wartung und Upgrades in die Erfassung integrieren
- den gesamten Lebenszyklus abbilden (von der Beschaffung bis zur Ausmusterung oder Rückgabe bei gemieteten Objekten)
- idealerweise auch die Wechselbeziehungen zu weiteren Bereichen (unter anderem Auftragsmanagement, Buchführung und Vertragsmanagement, Configuration- und Change-Management, Risikomanagement) berücksichtigen.

Um diese komplexen Aufgaben erfüllen zu können, werden als Ersatz manueller Prozesse – wo möglich – entsprechende Werkzeuge zur Erfassung und Verwaltung sowie zur Integration automatischer Datenerfassung eingesetzt. Dies können zum Beispiel Tools für das Tracking fester Bestände, eine automatische Benachrichtigung bezüglich kritischer Nutzungsdaten, Web-Reporting oder auch Datenexport sein.

Die Sammlung sämtlicher relevanter IT-Asset-Daten erfolgt in einer IT-Asset-Datenbank, deren Vorteile Schaffung von Transparenz, einfache Kontrolle und vorausschauende Planung komplexer IT-Umgebungen sind. Eine gut gepflegte Datenbank liefert hierbei die Basis für Produktivitäts- und Qualitätssteigerungen.

Unter dem Begriff des Configuration Items werden nach IT Infrastructure Library (ITIL) sämtliche an wichtigen Geschäftsprozessen beteiligten Betriebsmittel gezählt. Beispiele hierfür stellen unter anderem PCs, Server oder auch Software dar. Da insbesondere auch diese Configuration Items, die sowohl eine technische als auch eine Business-Relevanz haben, zu den IT-Assets zählen [Sch15, gar], wird im Rahmen dieser Arbeit der Begriff des (IT-) Assets auch für die Kombination von IP-Adresse, eingesetztem Betriebssystem, installierter Software oder betriebener Dienste genutzt.

Beispiele, in denen von einer vollständigen Asset-Datenbank profitiert werden kann, stellen zum einen der IT-Helpdesk dar, der auf die in der Datenbank zur Verfügung gestellten Informationen zu System und Konfiguration zurückgreift. Zum anderen profitieren Entwickler, die neue Systeme ausrollen möchten, vom genauen Wissen über die vorhandene Infrastrukturlandschaft [Wö].

2.4. Sample-Generator

Ein Sample-Generator ermöglicht durch weitgehend automatisierte Techniken das Generieren von Proben (Samples). Bei diesen Proben handelt es sich um Flow-Records, die um Informationen über die eingesetzten Systeme (Betriebssystem, Dienst und Software) angereichert werden. Die so erstellten Beispieldaten können im Anschluss unter anderem für das Trainieren des FRF-Tools genutzt werden.

Im Rahmen dieser Arbeit bezeichnet ein Sample-Generator ein komplexes System aus verschiedenen Rechnern mit unterschiedlichen Betriebssystemen und Softwareständen. Auf den einzelnen Rechnern werden manuell oder automatisiert vorher bekannte Aktionen gestartet. Zu diesen Aktionen zählen beispielsweise:

- Abrufen der Liste verfügbarer Updates
- Herunterladen verschiedener Updates
- Installieren von Software
- Betrieb von Software
- Aufrufen von Dokumenten im Internet

Durch das Wissen über die ausgeführten Aktionen sowie den Zeitpunkt der Ausführung ist es möglich, die Daten der Flow-Records wie eingehend genannt, um die Informationen über genutztes Betriebssystem, eingesetzte Software sowie Patchstand zu erweitern. Diese so gewonnen Daten lassen sich im Anschluss für Vergleiche sowie heuristische Verfahren zur Detektion beziehungsweise Wiedererkennung unbekannter Datensätze nutzen.

3. Anforderungen an das Flow-Record-Fingerprinting-Tool

Das im Rahmen dieser Arbeit zu entwickelnde Konzept sowie der hierfür zu entwickelnde Prototyp für ein Flow-Record-Fingerprinting-Tool (FRF-Tool) müssen sich an den Bedürfnissen der späteren Anwender orientieren. Um diese Bedürfnisse zu ermitteln, werden in Kapitel 3.1 die drei in Kapitel 1 erwähnten Fallbeispiele mit entsprechend unterschiedlichen Zielen beispielhaft erläutert. Danach lassen sich im Anschluss deren jeweilige Anforderungen an die Erkennung von Übertragungsinhalten mit den daraus folgenden Rückschlüssen auf das eingesetzte Betriebssystem und die verwendete Software ermitteln. Hier erfolgt auch eine erste Gewichtung.

Im Kapitel 3.2 erfolgt eine Zusammenfassung aller in Kapitel 3.1 ermittelten Anforderungen. Diese Basisanforderungen werden erneut gewichtet, gruppiert und zu Gesamtanforderungen zusammengefasst. Die so gewonnenen Gesamtanforderungen dienen als Basisanforderungen für das zu erstellende FRF-Tool.

3.1. Herleitung durch Fallbeispiele

Um ein möglichst breites Anforderungsspektrum zu erhalten, werden folgende Gruppen ausgewählt:

- Körperschaften öffentlichen Rechts: Sie widmen sich im Auftrag des Staates beziehungsweise der Länder oder Kommunen der Erfüllung einzelner und aus gesellschaftlichen Funktionen ableitbaren Bedürfnissen. Dabei streben sie nicht unbedingt nach Gewinn, sondern in der Regel nach Kostendeckung. Da im universitären Umfeld hierzu insbesondere auch die Aufgaben einer funktionierenden IT-Infrastruktur gehören, bieten sich für diese Arbeit als Beispiel Hochschulrechenzentren mit entsprechender Größe an.

- Privatrechtliche Unternehmen im Allgemeinen: Hierbei geht es weniger um eine betriebswirtschaftliche Sicht, sondern vielmehr um die Abgrenzung zu öffentlich-rechtlichen Institutionen. Damit sind hier Unternehmen mit einer IT-Infrastruktur zu verstehen, deren Ziel es ist, unter dem Aspekt von Markt- und Kapitalrisiken Unternehmenserfolg zu erwirtschaften.

- Strafverfolgungsbehörden und Nachrichtendienste: Diese Nischenorganisationen gehören zu den öffentlich-rechtlichen Institutionen. Aufgrund der von ihnen zu erbringenden Leistungen ergibt sich ein spezielles Anforderungsprofil.

Eine ausführliche Erörterung der festgestellten Anforderungen erfolgt zunächst am Beispiel der Hochschulrechenzentren, während – um Wiederholungen zu vermeiden – die speziellen Anforderungen von Unternehmen im Allgemeinen und von Nachrichtendiensten als zusätzliche Ergänzung dienen. Die hergeleiteten Anforderungen werden nach funktionalen und nicht funktionalen Anforderungen differenziert und im Anschluss für jeden Sachverhalt diskutiert.

3.1.1. Hochschulrechenzentren

Hochschulrechenzentren sind zentrale Einrichtungen von Universitäten mit Schwerpunkt auf Planung, Bereitstellung und Betrieb einer leistungsfähigen Kommunikationsinfrastruktur sowie Rechnern und Spezialgeräten [LR01]. Diese Infrastruktur sowie weitere IT-Dienstleistungen werden Universitäten, Studierenden, Mitarbeitern und Partnern der Universitäten zur Verfügung gestellt.

Als Beispiel für ein Hochschulrechenzentrum wird im Rahmen dieser Arbeit das Leibniz-Rechenzentrum (LRZ) mit Sitz in Garching bei München herangezogen. Das LRZ erfüllt Aufgaben eines Hochschulrechenzentrums für die Ludwig-Maximilians-Universität (LMU), die Technische Universität München (TUM), die Bayerische Akademie der Wissenschaften (BAdW), die Hochschule München (HM) und die Hochschule Weihenstephan. Zusätzlich betreibt das LRZ Hochleistungsrechensysteme für alle bayerischen Hochschulen. Im Zusammenhang mit diesen Aufgaben leistet das LRZ auch Forschung auf dem Gebiet der Angewandten Informatik [LR01].

Funktionale Anforderungen

Das Leibniz-Rechenzentrum (LRZ) ist organisatorisch an der Bayerischen Akademie der Wissenschaften angegliedert und wird von der Kommission für Informatik beaufsichtigt. Diese Kommission wird aus Vertretern der beiden Münchner Hochschulen und der Bayerischen Akademie der Wissenschaften gebildet und bestimmt aus ihrer Mitte ein Direktorium [LR01]. Dies wird in Abbildung 3.1 dargestellt.

Das Münchner Wissenschaftsnetz (MWN), das die meisten Gebäude der angeschlossenen Teilnehmer verbindet, ermöglicht die Datenkommunikation untereinander sowie den Zugang zum Internet. Der Backbone und viele Teilnetze werden vom LRZ betrieben, lokale Netze innerhalb von Instituten werden meist von Institutspersonal betreut [LR].

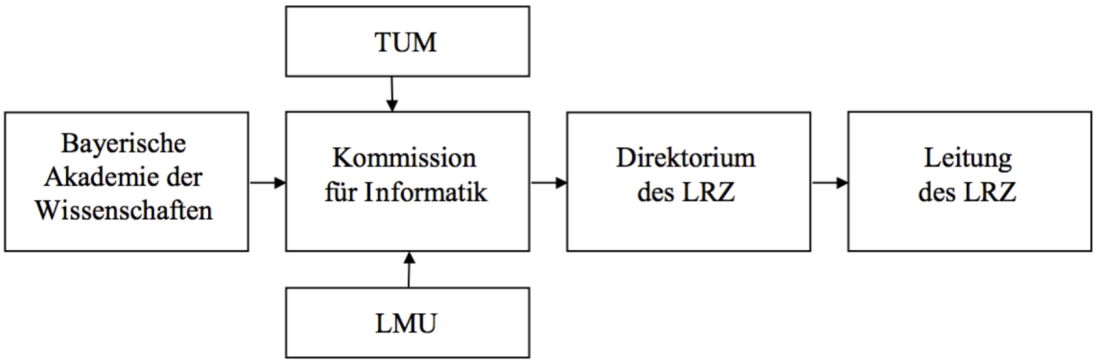


Abbildung 3.1.: Formale Ansiedlung des LRZ [LR01]

Funktionale Anforderung (FA) I

> *Die eingesetzte Lösung muss in heterogenen sowie eigenständigen Netzen erfolgreich einsetzbar sein.*

Die Größe des so entstandenen Rechnernetzes wird durch die schematische Darstellung der Backbones des Münchner Wissenschaftsnetz (MWN) in Abbildung 3.2 zusätzlich verdeutlicht.

Funktionale Anforderung (FA) II

> *Eine Erfassung und Analyse muss über (geographisch) weitläufige Netze möglich sein.*

Funktionale Anforderung (FA) III

> *Die Erfassung und Analyse muss in Border-Gateways zu autonomen Netzen möglich sein.*

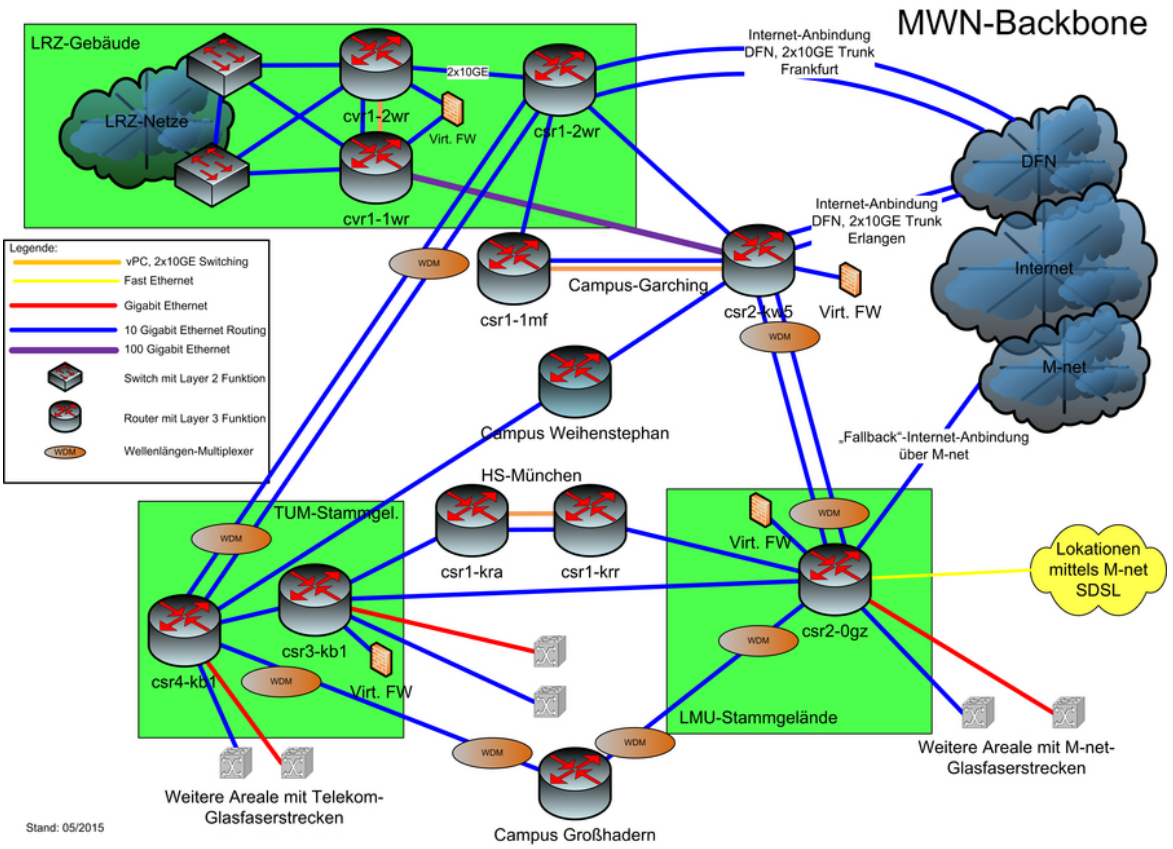


Abbildung 3.2.: Schematische Darstellung der Backbones des Müncher Wissenschaftsnetzes [LR16]

Anzahl der im LRZ betriebenen Server und Dienste

Das Leibniz-Rechenzentrum (LRZ) betrieb im hauseigenen Rechenzentrum im Jahr 2014 mehr als 2000 physikalische und virtuelle Server, von denen ein Teil auch an Dritte vermietet wurde [LR16]. Des Weiteren werden verschiedenste Dienste wie beispielsweise E-Mail, Exchange, Dateiaustausch, Webhosting oder Verzeichnisdienste zur Verfügung gestellt [LR15].

Funktionale Anforderung (FA) IV

Das Tool muss auch im Umfeld eines großen Rechenzentrums mit einer Vielzahl an Servern und hohem Datenaufkommen funktionieren.

Feststellen veralteter Software

Da eine manuelle Verwaltung einer derart hohen Anzahl an Servern aufwändig ist, bietet es sich an, die Konfiguration und Aktualisierung der Server und der dort betriebenen Dienste automatisiert durchzuführen. Doch auch bei Nutzung automatisierter Betriebssystem- oder Softwareaktualisierungen kann es zu Fehlern kommen. Tritt ein solcher Fehler auf, so ist es möglich, dass im Anschluss nach einem missglückten Update generell keine Aktualisierungen mehr auf dem Server installiert werden und damit Teile der installierten Software veraltet bleiben. In kleineren Rechnernetzen lassen sich fehlgeschlagene Updates beispielsweise durch E-Mail-Benachrichtigungen oder manuelle Kontrolle erkennen. Handelt es sich jedoch wie in diesem Beispiel um ein Rechenzentrum mit mehr als 2000 Servern, so wird ein automatisiertes System zur Erkennung des Ist-Standes benötigt.

Funktionale Anforderung (FA) V

Das Tool muss eingesetzte Betriebssysteme und Software inklusive deren Versionsstand erkennen.

Weitläufigkeit des MWN sowie Anzahl und Unabhängigkeit der Teilnehmer

Aus der Betrachtung des organisatorischen Aufbaus des LRZ und des MWN lässt sich erkennen, dass viele unabhängige Institutionen beziehungsweise Organisationen angebunden sind. So sind neben der LMU, der TUM sowie der weiteren Münchner Hochschulen auch das Studentenwerk München (StWM), dessen Wohnheime und weitere Organisationen an das MWN angeschlossen. Im Jahr 2014 erfolgte eine Erweiterung durch die Netzanbindung bayrischer Museen durch das LRZ an das MWN [LR15]. Die Graphik 3.3 verdeutlicht die geographische Ausdehnung des Münchner Wissenschaftsnetzes.

Neben den studentischen Anwendern des MWN nutzen auch Institute wie beispielsweise das Institut für Informatik (IfI) der LMU das MWN, um Assets für Mitarbeiter und Studenten bereitzustellen. Das Institut für Informatik (IfI) betreibt ferner ein eigenes Rechenzentrum, welches für die Studierenden des Institutes Dienste wie Datenbanken, Mailsystem oder auch verschiedene Versionskontrollsysteme zur Verfügung stellt [Ins]. Diese Systeme laufen unabhängig vom Leibniz-Rechenzentrum (LRZ) innerhalb des MWN, wobei das LRZ lediglich die IP-Adressen an das Rechenzentrum des IfI vergibt. Des Weiteren betreiben auch verschiedene Lehrstühle der Institute eigene Server oder Dienste, die dem Leibniz-Rechenzentrum (LRZ) nicht gemeldet werden müssen. Insbesondere der Betrieb von Honey-Pots, die der Erforschung der Ausbreitung von Schadsoftware dienen, sind für Netzverantwortliche von Interesse, da einer Ausbreitung dieser Schadsoftware innerhalb des ganzen Netzes vorgebeugt werden muss.

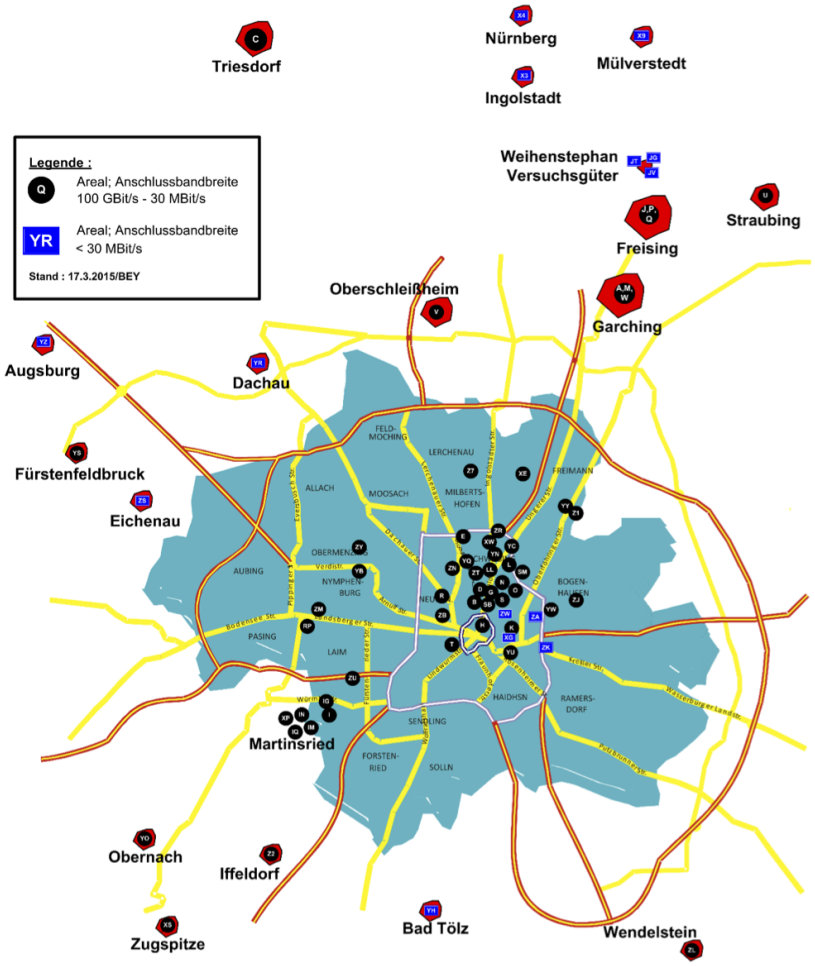


Abbildung 3.3.: Geographische Ausdehnung des MWN (nicht maßstabsgerecht) [LR15]

Funktionale Anforderung (FA) VI

> *Assets müssen auch erkannt werden, wenn das betroffene Netzsegment durch einen Dritten betreut wird.*

Die Betrachtung von Abbildung 3.3 verdeutlicht die Weitläufigkeit des MWN und damit die geographische Verteilung der unterschiedlichen Anschlusspunkte. Im Gegensatz zu einem Campus mit zentralem Knotenpunkt zur Datenanalyse, ist es im MWN erst im Backbone oder aber direkt in den angeschlossenen Gebäuden möglich, Daten zu erfassen. Durch Abbildung 3.2 wird ferner verdeutlicht, dass im MWN, analog zum Internet, mehrere alternative Pfade für den Datenverkehr existieren. Durch diesen Umstand ist nicht garantiert, dass bei einer Analyse „auf der Strecke“ alle notwendigen Datenpakete den Messpunkt durchlaufen, so dass die Datenerfassung nur am Anfang oder im zentralen Backbone möglich ist.

Funktionale Anforderung (FA) VII

> *Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.*

Das hohe Datenvolumen des MWN erschwert zusätzlich die Analyse. Bei einem Volumen im Petabytebereich, wobei dies lediglich der das Netz verlassende Traffic ist und der netzinterne Traffic noch nicht betrachtet wurde, ist eine vollständige Verkehrsanalyse durch Mirror-Ports mit nicht unerheblichen Kosten verbunden. Für eine derartige Trafficanalyse würde eine höhere Bandbreite sowie zusätzliche Hardware für Speicherung und Analyse benötigt.

Funktionale Anforderung (FA) VIII

Die Analyse muss bei hohem Trafficaufkommen zeitnah und zuverlässig einsetzbar sein und darf dabei keine Störungen verursachen.

Nicht Funktionale Anforderungen

Anzahl der im LRZ betriebenen Server und Dienste

Viele der mehr als 2000 im hauseigenen Rechenzentrum des LRZ betriebenen Server werden genutzt, um zentrale Dienste für das Münchner Wissenschaftsnetz (MWN) und seine Teilnehmer zur Verfügung zu stellen. Unter anderem stellt das LRZ Dienste wie Mail, Cloudspeicher, Webhosting und weitere hochverfügbare[1] Netzdienste bereit. Insbesondere die Verfügbarkeit von Authentifizierungs- sowie Autorisierungsservern oder auch Verzeichnisdiensten ist für Nutzer dieser Dienste von hoher Relevanz [LR15].

Nicht Funktionale Anforderung (NFA) I

Der Betrieb des Tools darf sich nicht negativ auf die Performance sowie Erreichbarkeit der betriebenen Server und Dienste auswirken. Überflüssige Logdateien oder die fälschliche Erkennung von Angriffen sind hierbei zu vermeiden.

Um einen einwandfreien Betrieb zu ermöglichen, darf ein derartiges Detektionssystem weder das Netz durch ein hohes Trafficaufkommen, wie es beispielsweise bei der passiven Detektion des gesamten Datenverkehrs entstehen kann, noch die Server durch eine Vielzahl an Logdateien oder durch Sicherheitssoftware fälschlicherweise erkannte Angriffe, die aus einer aktiven Detektion folgen können, belasten.

Nicht Funktionale Anforderung (NFA) II

Der Betrieb des Tools darf sich nicht negativ auf die Leistungsfähigkeit und Performance des betriebenen Netzes auswirken. Dies gilt insbesondere für eine Überlastung der Router durch eine Vielzahl an Anfragen sowie die Erzeugung von zusätzlichem Trafficaufkommen.

Ständig aktualisierte Asset-Datenbank

Aufgrund der hohen Anzahl der betriebenen Server ist es nur schwer möglich, ohne automatische Hilfsmittel einen Überblick über die vorhandenen Assets zu erhalten. Dabei ist es von großem Interesse, stets eine aktuelle Asset-Datenbank vorzuhalten.

Muss die Asset-Datenbank ganz oder teilweise von Hand gepflegt werden, so kann sich die erhebliche Anzahl der Mitarbeiter, die die mehr als 2000 Server betreuen, unter Umständen negativ auf die Vollständigkeit einer Asset-Datenbank auswirken.

[1] Hochverfügbarkeit (HA) ist eine Verfügbarkeitsklasse. Systeme und Netze, die mit Hochverfügbarkeit arbeiten, sind nach der Havard Research Group (HRG) zu 99,999 % verfügbar und haben eine jährliche Ausfallzeit von 8 Minuten. [ITW]

Nicht Funktionale Anforderung (NFA) III

> *Gewonnene Daten müssen für verschiedene Systeme lesbar und durch diese weiterverarbeitbar sein.*

Für den Fall, dass die Asset-Datenbank ein manuelles Eingreifen erfordert, besteht das Risiko darin, dass gegebenenfalls Änderungen nicht in einer bestehenden Asset-Datenbank aktualisiert oder nur mündlich innerhalb des Teams kommuniziert werden. Insbesondere Testsysteme oder temporär aufgesetzte Dienste werden erfahrungsgemäß häufig nicht in einer Asset-Datenbank erfasst. An dieser Stelle ist ein System wünschenswert, welches in Echtzeit eine Übersicht der vorhandenen Assets ausgeben und eine bestehende Datenbank um diese Informationen erweitern kann, ohne hierbei den Betrieb des Netzes zu stören oder negativ zu beeinflussen. Ferner ist es natürlich wünschenswert, die Kosten für ein derartiges System gering zu halten.

Nicht Funktionale Anforderung (NFA) IV

> *Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden.*

Traditionelle Analyseverfahren eignen sich in der Regel nicht für diese Aufgabe, da aktive Analyseverfahren Netzkomponenten überlasten oder Logfiles überfluten können. Passive Analyseverfahren eignen sich ebenfalls weniger durch den zusätzlich verursachten Traffic beziehungsweise die hohen Kosten durch Mirror-Ports und aufwändige Analyse des ausgeleiteten Datenstroms.

Nicht Funktionale Anforderung (NFA) V

> *Zur Erfassung der zu analysierenden Daten soll die bereits vorhandene Infrastruktur genutzt werden. Bereits vorhandene Möglichkeiten der Datenerfassung sind zu bevorzugen.*

Einführung des IPv6-Protokolls

Im Jahr 2014 wurde der Betrieb von IPv6 innerhalb des MWN weiter ausgebaut, so dass über 90.000 Endsysteme mit nativem IPv6 mit dem Netz verbunden waren [LR15]. In Zukunft ist davon auszugehen, dass die Anzahl der Systeme, welche natives IPv6 nutzen, weiter ansteigt und nicht nur IPv4 für den Datenverkehr genutzt wird.

Nicht Funktionale Anforderung (NFA) VI

> *Das zu entwickelnde Tool muss mit IPv6 kompatibel sein.*

Unabhängigkeit der Teilnehmer und Rechtssicherheit des Tools

Da es sich bei den Nutzern des MWN auch um vom LRZ unabhängige Dritte handelt, ergibt sich die Forderung nach einer rechtskonformen Lösung. Hierbei ist entscheidend, dass das Bundesdatenschutzgesetz sowie das Bayrische Datenschutzgesetz bei der Erhebung und Analyse der Daten eingehalten werden. Gerade personenbezogenen Daten gilt ein besonderes Augenmerk. Aber auch die Inhalte von Übertragungen, welche durch eine Deep-Packet-Inspection vollständig auswertbar wären, zählen zu den schützenswerten Daten. Besonders hervorzuheben ist an dieser Stelle die Einhaltung des §202a,b,c StGB (vgl. Anhang D). Unter Betrachtung des §303b StGB in Verbindung mit §202c StGB ist die Nutzung aktiver Netzscans über angeschlossene Geräte Dritter als kritisch einzustufen. Somit ist bei einem eingesetzten Analysetool unbedingt auf dessen Rechtskonformität zu achten.

Nicht Funktionale Anforderung (NFA) VII

Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.

Nicht Funktionale Anforderung (NFA) VIII

Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes sowie des Bayerischen Datenschutzgesetzes eingehalten werden. Dies gilt insbesondere für die Untersuchung von Paketinhalten.

Nicht Funktionale Anforderung (NFA) IX

Die Datenerfassung sowie die Analyse muss die Regelungen des StGB einhalten.

Zusammenfassung der Anforderungen

Aus den genannten Beispielen ist zu erkennen, dass sich insbesondere aus der Heterogenität und Unabhängigkeit der Teilnehmer Anforderungen ableiten lassen, die sich auf die im Netz befindlichen Assets beziehen. Bei der Erfassung dieser Assets ist es von Interesse, welche Betriebssysteme mit welchem Patchstand sowie welche Software in welcher Version eingesetzt werden. Wichtig ist hierbei, dass die bestehende Infrastruktur und die Informationssysteme Dritter durch ein Detektionstool nicht gestört oder überlastet werden. Rechtskonformität und geringe Kosten runden die Anforderungen ab.

Zusammengefasst benötigt ein Hochschulrechenzentrum ein Hilfsmittel, um ohne negative Beeinflussung des Netzes sowie der Netzteilnehmer in Hinblick auf Performance oder Verzögerungen eine aktuelle Asset-Datenbank zu erstellen. Hierbei soll die Asset-Datenbank den Patch-Stand der Betriebssysteme sowie der dort installierten Software widerspiegeln.

Die so abgeleiteten funktionalen Anforderungen werden in Tabelle 3.1 zusammen mit einem Schlagwort sowie einer nachfolgend erläuterten persönlichen Priorisierung der Anforderung aufgeführt.

- FA I: Die Einsetzbarkeit der entwickelten Lösung in heterogenen Netzen ist für das LRZ auf Grund des Aufbaus des Netzes von sehr hoher Wichtigkeit. (+++)
- FA II: Die Möglichkeit, mit der entwickelten Lösung in (geographisch) weitläufigen Netzen Daten erfassen und analysieren zu können, ist auf Grund der Größe des MWN für das LRZ von hoher Wichtigkeit. (++)
- FA III: Die Möglichkeit, Daten an Grenzpunkten zu autonomen Netzen erfassen zu können, ist für das LRZ, so lange es möglich ist qualitative Daten anderweitig erfassen zu können, nicht von großer Wichtigkeit. (+)
- FA IV: Die Nutzbarkeit des Tools im Umfeld großer Rechenzentren mit vielen Servern und hohem Trafficaufkommen ist für das LRZ auf Grund der Anzahl der betriebenen Server und des anfallenden Datenvolumens von sehr hoher Wichtigkeit. (+++)
- FA V: Da die Erkennung vorhandener Assets den Hauptzweck des Tools darstellt, ist diese Anforderung von besonderer Wichtigkeit. (+++)
- FA VI: Da das MWN viele autonome Teilnetze besitzt, ist es notwendig und wichtig, die sich im Netz befindlichen Assets auch erkennen zu können. (++)

- FA VII: Da im MWN mehrere alternative Routen existieren, über die Daten geleitet werden können, ist es notwendig mit dynamischem Routing kompatibel zu sein. Da dies jedoch durch geeignete Erfassungsorte möglich ist, kann diese Anforderung auch vernachlässigt werden. (+)
- FA VIII: Die störungsfreie Bereitstellung des MWN sowie der betriebenen Dienste ist für das LRZ existenziell wichtig. Da das im MWN auftretende Datenvolumen von nicht unerheblicher Größe ist, ist diese Anforderung sehr wichtig. (+++)

Tabelle 3.1.: Funktionale Anforderungen aus Sicht eines Hochschulrechenzentrums

ID	**Schlagwort**	**Anforderung**	**Priorität**
FA I	Heterogene Netze	Die eingesetzte Lösung muss in heterogenen sowie eigenständigen Netzen erfolgreich einsetzbar sein.	+++
FA II	Weitläufigkeit	Eine Erfassung und Analyse muss über (geographisch) weitläufige Netze möglich sein.	++
FA III	Weitläufige Netze	Die Erfassung und Analyse muss in Border-Gateways zu autonomen Netzen möglich sein.	+
FA IV	Trafficaufkommen	Das Tool muss auch im Umfeld eines großen Rechenzentrums mit einer Vielzahl an Servern und hohem Datenaufkommen funktionieren.	+++
FA V	Asseterkennung	Das Tool muss eingesetzte Betriebssysteme und Software inklusive deren Versionsstand erkennen.	+++
FA VI	Fremdbetreute Netze	Assets müssen auch erkannt werden, wenn das betroffene Netzsegment durch einen Dritten betreut wird.	++
FA VII	Routing	Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.	+
FA VIII	Störungsfreiheit	Die Analyse muss bei hohem Trafficaufkommen zeitnah und zuverlässig einsetzbar sein und darf dabei keine Störungen verursachen.	+++

\+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

In Tabelle 3.2 erfolgt eine vergleichbare Aufstellung für Nicht Funktionale Anforderungen. Nachfolgend werden die vorhergehend beschriebenen Anforderungen kurz diskutiert und deren Priorität festgelegt:

- NFA I: Die störungsfreie Bereitstellung des MWN sowie der dort betriebenen Dienste stellt eine der zentralen Aufgaben des LRZ dar. Daher ist es von besonderer Wichtigkeit, jedwede Störung zu vermeiden. (+++)
- NFA II: Wie bereits in NFA I beschrieben stellt die störungsfreie Bereitstellung des MWN sowie der dort betriebenen Dienste eine der zentralen Aufgaben des LRZ dar. Insbesondere durch die hohe Anzahl an teilnehmenden Geräten und Nutzern ist es von besonderer Wichtigkeit, jede Störung der Netzhardware sowie der Performance zu vermeiden. (+++)
- NFA III: Gewonnene Daten müssen durch weitere eingesetzte Software nutzbar sein, jedoch ist es auch möglich auf der Gegenseite entsprechende Anpassungen zum Einlesen der gewonnenen Daten durchzuführen. Daher ist diese Anforderung für ein zu entwickelndes Tool nicht zwingend notwendig. (+)
- NFA IV: Die Vermeidung von zusätzlichen Kosten ist zwar erstrebenswert, jedoch kann diese Anforderung, wenn kein anderer Weg möglich ist, vernachlässigt werden. (+)
- NFA V: Um unnötigen Aufwand für Installation sowie Beschaffung zu vermeiden, sollte bestehende Hardware genutzt werden. Ist dies jedoch nicht möglich, so kann diese Anforderung ebenfalls vernachlässigt werden. (+)
- NFA VI: Um zukunftskompatibel zu sein, sollte IPv6 unterstützt werden. Auch wenn ein Großteil der Teilnehmer noch IPv4 nutzt, so muss strategisch von einem Einsatz von IPv6 ausgegangen werden. (++)
- NFA VII: Um den Datenschutz gewährleisten zu können, ist es notwendig, die gesammelten Daten zu anonymisieren. Daher ist diese Anforderung von besonderer Wichtigkeit. (+++)
- NFA VIII: Die Einhaltung des Datenschutzes auf Grund gesetzlicher Anforderungen und Regelungen ist für eine öffentlich rechtliche Institution von sehr hoher Wichtigkeit. (+++)
- NFA IX: Die Einhaltung gesetzlicher Anforderungen und Regelungen ist für eine öffentlich rechtliche Institution von sehr hoher Wichtigkeit. (+++)

Tabelle 3.2.: Nicht Funktionale Anforderungen aus Sicht eines Hochschulrechenzentrums

ID	**Schlagwort**	**Anforderung**	**Priorität**
NFA I	Dienstperformance	Der Betrieb des Tools darf sich nicht negativ auf die Performance sowie Erreichbarkeit der betriebenen Server und Dienste auswirken. Überflüssige Logdateien oder die fälschliche Erkennung von Angriffen sind hierbei zu vermeiden.	+++
NFA II	Netzperformance	Der Betrieb des Tools darf sich nicht negativ auf die Leistungsfähigkeit und Performance des betriebenen Netzes auswirken. Dies gilt insbesondere für eine Überlastung der Router durch eine Vielzahl an Anfragen sowie die Erzeugung von zusätzlichem Trafficaufkommen.	+++
NFA III	Weiterverarbeitbarkeit	Gewonnene Daten müssen für verschiedene Systeme lesbar und durch diese weiterverarbeitbar sein.	+
NFA IV	Kosteneffizienz	Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden.	+
NFA V	Bestehende Infrastruktur	Zur Erfassung der zu analysierenden Daten soll die bereits vorhandene Infrastruktur genutzt werden. Bereits vorhandene Möglichkeiten der Datenerfassung sind zu bevorzugen.	+
NFA VI	IPv6	Das zu entwickelnde Tool muss mit IPv6 kompatibel sein.	++
NFA VII	Anonymisierung	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	+++
NFA VIII	Datenschutz	Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes sowie des Bayerischen Datenschutzgesetzes eingehalten werden. Dies gilt insbesondere für die Untersuchung von Paketinhalten.	+++
NFA IX	Gesetzeskonformität	Die Datenerfassung sowie die Analyse muss die Regelungen des StGB einhalten.	+++

+ Anforderung nice to have/ ++ Anforderung wichtig / +++ Pflichtanforderung

3.1.2. Unternehmen

Unternehmen aus der Sicht dieser Arbeit besitzen eine IT-Infrastruktur und haben letztendlich ähnliche Anforderungen wie unter 3.1.1 genannt. Auch hier sollen im Hinblick auf das Netz möglichst keine negativen Beeinflussungen entstehen. Die Heterogenität und Unabhängigkeit ist zwar nicht im gleichen Umfang wie im Beispiel des LRZ vorhanden, dafür gibt es aber unabhängige Fachabteilungen, Tochtergesellschaften oder beteiligte Unternehmen, die zu berücksichtigen sind. Für die verantwortliche IT-Abteilung ist daher auch hier das Ziel, eine einwandfrei funktionierende IT-Landschaft bereitzustellen.

Funktionale Anforderungen

In diesem Punkt soll explizit auf die Existenz einer Schatten-IT in Unternehmen eingegangen werden. Diese kann sich durch sämtliche Branchen ziehen.

Eine Schatten-IT beschreibt informationstechnische System-, Prozess- und Organisationseinheiten, die in den Fachabteilungen neben der offiziellen IT-Infrastruktur sowie ohne das Wissen der IT-Abteilung existieren (Zweitsystem). Somit ist diese nicht in die IT-Infrastruktur eingebunden und unterliegt daher nicht dem IT-Management bezüglich Freigabe und Aktualisierung. Häufig ist die Schatten-IT sehr aufgabenorientiert und auf interne Prozesse von Fachabteilungen fokussiert. Damit steht sie oftmals nur einem bestimmten beziehungsweise eingeschränkten Personenkreis zur Verfügung. Die Entstehung von Schatten-IT kann in verschiedenen Ursachen begründet sein. Hierzu zählen unter anderem:

- Langwierige Genehmigungs- und Planungsverfahren durch die jeweilige IT-Abteilung sowie unangebrachte Koordinationsmechanismen
- Personelle oder finanzielle Ressourcen-Engpässe in der IT-Abteilung durch fehlendes Know-How oder zu starre Budgets
- Möglichkeit des Zugangs zu Anwendungen außerhalb des IT-Bereichs und leichte Anschaffungen von IT-Lösungen direkt in den Fachabteilungen; teilweise können auch eigene Geräte für die Arbeit genutzt werden (bring your own device). Die Nutzung von Kleinst-Computern wie Raspberry PI erlaubt ebenfalls den Betrieb eigener Dienste am Arbeitsplatz.
- Vertrautheit mit bestimmten Anwendungen aus dem Privatbereich (Dropbox, Owncloud, etc.)
- Mangelnde Abgrenzung von Verantwortlichkeiten innerhalb der Abteilungen und schlechte organisatorische Abstimmung insbesondere zwischen Fachabteilung und IT
- Zugriff von Dritten auf das interne Rechnernetz wie beispielsweise beauftragte externe Berater, die sich mit eigenen Geräten im Unternehmensnetz aufhalten

Das Ziel des Unternehmens liegt oftmals nicht darin, die Schatten-IT abzuschaffen, sondern diese innerhalb der IT-Infrastruktur nutzergetrieben weiterzuentwickeln und in ein offizielles System zu überführen.

Funktionale Anforderung (FA) IX

Die eingesetzte Lösung muss in heterogenen sowie konzernübergreifenden Netzen einsetzbar sein.

Funktionale Anforderung (FA) X

Eine Erfassung und Analyse muss standortübergreifend und weltweit möglich sein.

Funktionale Anforderung (FA) XI

Im Netz vorhandene Assets müssen vollständig und korrekt erkannt werden.

Dies ist allerdings nicht als Momentaufnahme, sondern als kontinuierlicher Prozess zu betrachten. Nach der Aufnahme der betreffenden Anwendungen müssen diese geprüft, ins offizielle IT-Management integriert oder unterbunden werden, so dass am Ende das offizielle Angebot der IT verbessert wird, weil es sich am tatsächlichen Bedarf orientiert.

Funktionale Anforderung (FA) XII

Die Analyse muss zeitnah und zuverlässig durchführbar sein.

Für Administratoren nimmt die Komplexität der Betreuung der Infrastruktur mit der Expansion von Unternehmen weiter zu. In Unternehmen mit nur wenigen Mitarbeitern ist es für IT-Verantwortliche noch möglich, vorhandene Assets von Hand durch Protokollierung, manuelle Auswertung oder Analyse von Log-Dateien zu verwalten. Mit einer zunehmenden Anzahl an Mitarbeitern und der damit verbundenen räumlichen Ausdehnung wie auch der Zunahme an Datendurchsatz steigt in der Regel auch die Anzahl der betriebenen Dienste.

Funktionale Anforderung (FA) XIII

Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein.

Funktionale Anforderung (FA) XIV

Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.

Nicht Funktionale Anforderungen

Eine erhebliche Schwierigkeit beim Vorhalten einer ständig aktualisierten Asset-Datenbank ist in der geringen Anzahl der auf dem Markt befindlichen qualifizierten Analyse- und Auswertungstools zu finden. Für einen reibungslosen Geschäftsablauf ist es für Unternehmen wichtig, Kontrolle über die eigenen Daten zu besitzen sowie den reibungsfreien Zugang für berechtigte Mitarbeiter zu ermöglichen. In diesem Sinne sollen unter anderem durch Mitarbeiter erstellte Umgehungslösungen wie eigene Dateiablagen, Fileshares wie Owncloud sowie selbst betriebene Dienste, die zur Schatten-IT zählen, erkannt werden.

Nicht Funktionale Anforderung (NFA) X

Eine Analyse muss während des Tagesgeschäftes möglich sein und die Teilnehmer dürfen dabei nicht gestört oder behindert werden. Die Analyse soll transparent für den Anwender stattfinden.

Nicht Funktionale Anforderung (NFA) XI

Die Infrastruktur darf durch das Tool nicht beeinträchtigt werden (Überlastung der Router, zu viele Logdateien).

Nach Auswertung des Netzes durch ein geeignetes Tool ist es für Administratoren auch von Relevanz, Änderungen im Netz erfassen und einen geprüften Ist-Stand ablegen zu können. Insbesondere Wegfall und Hinzukommen von Diensten oder ganzen Hosts sind hierbei von besonderer Wichtigkeit und sollten gegebenenfalls aktiv an Verantwortliche weitergegeben werden können. Eine Einbindung des Tools in bestehende Monitoring-Werkzeuge wie Nagios oder auch die Zusammenführung der Ergebnisse bestehender Überwachungs- sowie Monitoring-Tools ist daher wünschenswert.

Nicht Funktionale Anforderung (NFA) XII

Das Tool muss maschinenlesbare Ausgaben in einem festgelegten Format liefern.

Nicht Funktionale Anforderung (NFA) XIII

Eine Integration durch offene Schnittstellen muss möglich sein.

Für Unternehmen ist es wichtig, die geltenden gesetzlichen Regelungen und Anforderungen einzuhalten. Hierzu zählen verschiedene Gesetze wie beispielsweise das Bundesdatenschutzgesetz. Die Einhaltung des Datenschutzes innerhalb des Firmenumfelds wird in der Regel durch den Datenschutzbeauftragten der jeweiligen Unternehmung überwacht.

Nicht Funktionale Anforderung (NFA) XIV

Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.

Nicht Funktionale Anforderung (NFA) XV

Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes (BDSG) sowie des Bayerischen Datenschutzgesetzes (BayDSG) eingehalten werden.

Zusammenfassung der Anforderungen

Kurz gesagt liegen damit die Anforderungen von Unternehmen an ein geeignetes Werkzeug in der Unterstützung zum Erhalt eines aktuellen Ist-Standes und damit einer aktuellen Asset-Datenbank, die insbesondere die Aktualisierung der IT-Landschaft berücksichtigt und die zukünftige Planung erleichtert. Selbstverständlich darf es insbesondere während der regelmäßigen Arbeitszeiten zu keinen Netzstörungen kommen und da es sich um gewinnorientierte Unternehmen handelt, muss die Lösung in einem angemessenen Kosten/Nutzen-Verhältnis stehen. Betrachtet man diese Anforderungen ergeben sich folgende Prioritäten für die Funktionalen Anforderungen:

- FA IX: Da besonders große Unternehmen mit weitläufigen Netzen eine Zielgruppe des Tools sind, ist die Kompatibilität mit firmenübergreifenden Netzen von hoher Priorität. (++)
- FA X: Da besonders große Unternehmen mit (geographisch) weitläufigen Netzen eine Zielgruppe des Tools sind, ist die Kompatibilität mit weitläufigen Netzen wichtig. (++)

- FA XI: Da die Erkennung der im Netz vorhandenen Assets die Hauptfunktion des Tools darstellt, ist diese Anforderung von sehr hoher Priorität. (+++)
- FA XII: Um eine Analyse – während der sich die Geräte eingeschaltet im Netz befinden – durchführen zu können, ist es notwendig, zeitnah eine Auswertung erstellen zu können. Da sich Analyse und Datenerfassung jedoch trennen lassen, kann eine verzögerte Analyse ebenfalls zum Ziel führen. Somit ist die Anforderung vernachlässigbar. (+)
- FA XIII: Da große Unternehmensnetze oft entsprechend viele Teilnehmer mit hohem Datendurchsatz haben, ist diese Anforderung als sehr wichtig einzustufen. (+++)
- FA XIV: Die Analyse innerhalb von Netzen mit dynamischem sowie komplexem Routing lässt sich durch geeignete Wahl der Überwachungsknoten optimieren, so dass diese Anforderung vernachlässigt werden kann. (+)

Tabelle 3.3.: Funktionale Anforderungen aus Sicht eines Unternehmens

ID	**Schlagwort**	**Anforderung**	**Priorität**
FA IX	Netzübergreifend	Die eingesetzte Lösung muss in heterogenen sowie konzernübergreifenden Netzen einsetzbar sein.	++
FA X	Weitläufigkeit	Eine Erfassung und Analyse muss standortübergreifend und weltweit möglich sein.	++
FA XI	Asseterkennung	Im Netz vorhandene Assets müssen vollständig und korrekt erkannt werden.	+++
FA XII	Zeitnähe	Die Analyse muss zeitnah und zuverlässig durchführbar sein.	+
FA XIII	Trafficaufkommen	Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein.	+++
FA XIV	Routing	Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.	+

\+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

Für die Nicht Funktionalen Anforderungen ergeben sich durch nachfolgende Diskussion die anschließenden Prioritäten:

- NFA X: Auf Grund der Arbeitszeiten der Mitarbeiter (Betrieb von Schatten-IT, verbundene Systeme sind eingeschaltet, IT-Abteilung ist anwesend) ist es wichtig, während der regulären Arbeitszeit Analysen durchführen zu können. Die Vermeidung einer Störung der Mitarbeiter wie auch die Reduzierung von deren Leistungsfähigkeit ist auf Grund möglicher Kosten oder Verluste zu berücksichtigen. Aus diesem Grund ist diese Anforderung von besonderer Wichtigkeit. (+++)

- NFA XI: Weder Mitarbeiter bei ihrer Arbeit zu behindern noch Netzausfälle durch die Analyse zu verursachen, muss hoch priorisiert werden, da Ausfälle oft auch mit weiteren Kosten verbunden sind. (+++)
- NFA XII: Für eine automatisierte Weiterverarbeitung der gewonnenen Informationen ist es notwendig, diese in festgelegter maschinenlesbarer Form vorzuhalten. Durch clientseitige Erstellung geeigneter Parser kann diese Anforderung jedoch vernachlässigt beziehungsweise aufgeschoben werden. (+)
- NFA XIII: Die Integration des Tools in offene Schnittstellen sowie das Bereitstellen offener Schnittstellen kann auf eine spätere Version nachgelagert werden. Die Kommunikation kann ferner auch über die Analyseausgabe erfolgen, so dass die Anforderung vernachlässigt werden kann. (+)
- NFA XIV: Um personenbezogene Daten speichern und auswerten zu dürfen, ist der Grundsatz der Datensparsamkeit unbedingt einzuhalten. Hiermit ist eine geeignete Anonymisierung der gespeicherten Daten äußerst wichtig. (+++)
- NFA XV: Da insbesondere für die automatische Auswertung von Personendaten besondere Schutzvorkehrungen zu treffen sind, ist die Einhaltung des Datenschutzes von besonderer Wichtigkeit. (+++)

Tabelle 3.4.: Nicht Funktionale Anforderungen aus Sicht eines Unternehmens

ID	**Schlagwort**	**Anforderung**	**Priorität**
NFA X	Störungsvermeidung	Eine Analyse muss während des Tagesgeschäftes möglich sein und die Teilnehmer dürfen dabei nicht gestört oder behindert werden. Die Analyse soll transparent[2] für den Anwender stattfinden.	+++
NFA XI	Ausfallfreiheit	Die Infrastruktur darf durch das Tool nicht beeinträchtigt werden (Überlastung Router, zu viele Logdateien).	+++
NFA XII	Maschinenlesbarkeit	Das Tool muss maschinenlesbare Ausgaben in einem festgelegten Format liefern.	+
NFA XIII	Schnittstellen	Eine Integration durch offene Schnittstellen muss möglich sein.	+
NFA XIV	Anonymisierung	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	+++
NFA XV	Datenschutz	Bei der Datenerfassung muss der Datenschutz im Sinne des BDSG sowie des BayDSG eingehalten werden.	+++

+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

[2]Unter Transparenz versteht man, dass ein bestimmter Teil eines Systems zwar in Betrieb ist, vom Benutzer aber nicht als vorhanden wahrgenommen werden kann und für diesen daher unsichtbar ist.

3.1.3. Strafverfolgungsbehörden und Nachrichtendienste

Strafverfolgungsbehörden und Nachrichtendienste besitzen die gleiche Basisaufgabe, nämlich die der Informationsbeschaffung. Dabei versteht man unter Strafverfolgung im engeren Sinne die Tätigkeit der Staatsanwaltschaft bis zur Anklageerhebung, wobei als Voraussetzung für notwendige Ermittlungen ein begründeter Verdacht vorliegen muss, der zur Erhebung von Beweisen führen soll [Los07, Sta]. Damit muss die Polizei im Gegensatz zu den Nachrichtendiensten auf die Aufklärung rechtswidriger Handlungen und Zustände beschränkt bleiben, so dass im Falle von legalem Verhalten nicht ermittelt werden darf.

Funktionale Anforderungen

Nachrichtendienste oder auch Geheimdienste agieren meist in Form einer Behörde und dabei durchaus auch ohne Anfangsverdacht. Sie sind prinzipiell von Strafverfolgungsbehörden unabhängig. Die Sammlung und Auswertung von Informationen erfolgt dabei mit nachrichtendienstlichen Mitteln, wobei diese insbesondere Methoden, Gegenstände und Instrumente der heimlichen Informationsbeschaffung umfassen [BND, Bun].

Funktionale Anforderung (FA) XV

Das Tool muss genaue Rückschlüsse auf Übertragungsinhalte ermöglichen.

Funktionale Anforderung (FA) XVI

Die Datenerfassung muss am Netzabschlusspunkt möglich sein.

Die Erfassung und Sammlung von Informationen kann sowohl bei Strafverfolgungsbehörden als auch bei Nachrichtendiensten an unterschiedlichen Orten erfolgen. Werden Informationen über Unternehmen oder Organisationen gesammelt, so ist es oft notwendig Daten außerhalb, also an Grenzpunkten zum Netz, zu sammeln. Bei einer Überwachung von Einzelpersonen ist es möglich, die Daten entweder nah am Netzabschlusspunkt der zu überwachenden Person oder aber an einer zentralen Stelle zu erfassen. Unabhängig von der Struktur des Netzes oder Art und Umfang der angeschlossen Systeme muss ein Flow-Record-Fingerprinting-Tool (FRF-Tool) vollumfänglich einsetzbar sein.

Funktionale Anforderung (FA) XVII

Das Tool muss in heterogenen Systemen einsetzbar sein.

Funktionale Anforderung (FA) XVIII

Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein.

Funktionale Anforderung (FA) XIX

Das Tool muss Assets auch in durch Dritte betreuten Netzen erkennen können.

Nicht Funktionale Anforderungen

Wie in der Publikation „Spionage Ihre Ziele Ihre Methoden“ des Bundesamtes für Verfassungsschutz nachzulesen ist, muss die Informationsbeschaffung so erfolgen, dass der Betroffene dies nicht bemerkt [Bun14]. Diese passive Überwachung wird durch methodische Vorgehensweisen unterstützt, wozu insbesondere die Überwachung und Ausspähung von Kommunikationsverbindungen (Telefonie und Internetverkehr) sowie elektronische Angriffe auf IT-Infrastrukturen zu nennen sind.

Nicht Funktionale Anforderung (NFA) XVI

Die Datenerfassung muss auch mit modernen Protokollen (IPv6) kompatibel sein.

Nicht Funktionale Anforderung (NFA) XVII

Ein mögliches Tool muss für die überwachten Teilnehmer transparent sein.

Nachrichtendienste unterliegen bei der Telekommunikationsüberwachung (TKÜ) vielen nationalen gesetzlichen Bestimmungen und Richtlinien, von denen nur einige wie Strafgesetz, Strafprozessordnung oder BKA-Gesetz genannt werden sollen. Darunter gibt es auch eine technische Richtlinie zur Umsetzung der gesetzlichen Maßnahmen zur Überwachung der Telekommunikation und zum Auskunftsersuchen für Verkehrsdaten (TRTKÜV), in welcher unter anderem detailliert die Überwachung der Verbindungen auf leitungsvermittelnden Netzen, die Überwachung von Speichereinrichtungen, die Überwachung des E-Mail-Dienstes und die Anforderungen für den Internetzugangsweg erläutert werden [Wei11].

Nicht Funktionale Anforderung (NFA) XVIII

Eine gezielte Auswahl der zu analysierenden Endpunkte muss möglich sein.

Nicht Funktionale Anforderung (NFA) XIX

Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.

Nicht Funktionale Anforderung (NFA) XX

Bei der Datenerfassung muss der Datenschutz im Sinne des BDSG sowie des BayDSG eingehalten werden. Dies gilt insbesondere für die Untersuchung von Paketinhalten.

Nicht Funktionale Anforderung (NFA) XXI

Die Datenerfassung sowie die Analyse muss die Regelungen des StGB einhalten.

Zusammenfassung der Anforderungen

Betrachtet man diesen hier beschriebenen Sonderfall näher, so lassen sich die Besonderheiten an die Anforderungen für das zu erstellende Tool wie folgt definieren: Dieses Werkzeug muss in kürzester Zeit unbemerkt und so genau wie möglich Computermerkmale identifizieren, die zu möglichst exakten Ergebnissen hinsichtlich der übertragenen Inhalte sowie der eingesetzten Software führen. Entsprechend lassen sich für die funktionalen Anforderungen die folgenden Priorisierungen herleiten:

- FA XV : Die Detektion der Übertragungsinhalte stellt ein Hauptziel bei der Überwachung durch Strafverfolgungsbehörden sowie Nachrichtendienste dar und ist daher von sehr hoher Wichtigkeit. (+++)
- FA XVI : Um eine Überwachung unbemerkt für den Überwachten durchführen zu können, ist oftmals eine gewisse Distanz notwendig. Aus diesem Grund ist es sehr wichtig, am Netzabschlusspunkt oder weiter entfernt qualitative Analysen durchführen zu können. (+++)
- FA XVII : Da überwachte Netze und Rechner oft verschiedene Konfigurationen besitzen und aus unterschiedlicher Hardware bestehen, ist es von Wichtigkeit auch mit heterogenen Systemen kompatibel zu sein. (++)
- FA XVIII : Werden Unternehmen oder größere Organisationen überwacht, so ist es notwendig, auch in weitläufigen Netzen eine Analyse durchführen zu können. Da Strafverfolgungsbehörden und Nachrichtendienste in der Regel sehr zielgerichtet arbeiten, kann diese Anforderung vernachlässigt werden. (+)
- FA XIX : Da Nachrichtendiensten bei Spionage und Überwachung oftmals die Kontrolle über das überwachte Netzsegment fehlt, ist die Möglichkeit auch in Netzen Dritter eine Analyse durchführen zu können von Wichtigkeit. (++)

Tabelle 3.5.: Funktionale Anforderungen aus Sicht von Strafverfolgungsbehörden und Nachrichtendiensten

ID	**Schlagwort**	**Anforderung**	**Priorität**
FA XV	Inhaltsdetektion	Das Tool muss genaue Rückschlüsse auf Übertragungsinhalte ermöglichen.	+++
FA XVI	Netzabschluss-punkt	Die Datenerfassung muss am Netzabschlusspunkt möglich sein.	+++
FA XVII	heterogene Systeme	Das Tool muss in heterogenen Systemen einsetzbar sein.	++
FA XVIII	Weitläufigkeit	Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein.	+
FA XIX	Netze Dritter	Das Tool muss Assets auch in durch Dritte betreuten Netzen erkennen können.	++

+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

Für die Nicht Funktionalen Anforderungen seitens Strafverfolgungsbehörden und Nachrichtendiensten ergeben sich folgende Prioritäten:

- NFA XVI: Die Anforderung, neue Übertragungsprotokolle wie IPv6 zu unterstützen, ist auf Grund der steigenden Verbreitung dieses Protokolls für Strafverfolgungsbehörden und Nachrichtendienste, welche nach Möglichkeit die vollständigen Informationen sammeln möchten, von sehr hoher Wichtigkeit. (+++)

- NFA XVII: Da Strafverfolgungsbehörden und Nachrichtendienste oft im Geheimen agieren, muss das Tool – um keine Aufmerksamkeit zu erwecken – unbedingt transparent für die Zielperson sein. (+++)
- NFA XVIII: Auf Grund gesetzlicher Vorgaben ist es für Strafverfolgungsbehörden und Nachrichtendienste notwendig, die Überwachung möglichst präzise einschränken zu können, um lediglich die Grundrechte der angeordneten Zielperson/en einzuschränken. Daher ist die Möglichkeit gezielt zu selektieren von besonderer Wichtigkeit. (+++)
- NFA XIX: Für die Einhaltung des gesetzlichen Datenschutzes ist es zwingend notwendig, Interaktionen mit Dritten soweit möglich geeignet zu anonymisieren. (+++)
- NFA XX: Werden Grundrechte von Bürgern beschnitten, so ist es notwendig, den Datenschutz wie auch weitere gesetzliche Anforderungen zu erfüllen. Insbesondere wenn Regelungen des Fernmeldegesetzes durch Untersuchung von Kommunikation eingeschränkt werden, ist es besonders wichtig, die grundlegenden Anforderungen des Datenschutzes einzuhalten. (+++)
- NFA XXI: Insbesondere für Strafverfolgungsbehörden und Nachrichtendiensten, welche in der Öffentlichkeit ein gewisses Vertrauen besitzen, ist es dringend notwendig, gesetzliche Regelungen einzuhalten. (+++)

Tabelle 3.6.: Nicht Funktionale Anforderungen aus Sicht von Strafverfolgungsbehörden und Nachrichtendiensten

ID	**Schlagwort**	**Anforderung**	**Priorität**
NFA XVI	IPv6	Die Datenerfassung muss auch mit modernen Protokollen (IPv6) kompatibel sein.	+++
NFA XVII	Transparenz	Ein mögliches Tool muss für die überwachten Teilnehmer transparent sein.	+++
NFA XVIII	Zielgerichtet	Eine gezielte Auswahl der zu analysierenden Endpunkte muss möglich sein.	+++
NFA XIX	Anonymisierung	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	+++
NFA XX	Datenschutz	Bei der Datenerfassung muss der Datenschutz im Sinne des BDSG sowie des BayDSG eingehalten werden. Dies gilt insbesondere für die Untersuchung von Paketinhalten.	+++
NFA XXI	Gesetzeskonformität	Die Datenerfassung sowie die Analyse muss die Regelungen des StGB einhalten.	+++

\+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

3.2. Diskussion und Zusammenfassung

Im Kapitel 3.1. wurden unterschiedliche Beispiele erläutert, die zu unterschiedlichen Anforderungsprofilen an ein FRF-Tool führen. Während ein Hochschulrechenzentrum dieses Tool nutzen kann, um eine möglichst aussagefähige Asset-Datenbank der selbst betriebenen Server und Dienste zu generieren, benötigt der Betreiber eines Hochschulnetzes zusätzlich ein Tool, welches das Netz nicht beeinträchtigt und mit Rechtskonformität sowie geringen Kosten verbunden ist.

Unternehmen unterliegen im Rahmen der Ergebnisorientierung verstärkt einem gewissen Kosten-Nutzen-Verhältnis und benötigen ebenfalls störungsfreie Netze sowie eine gepflegte und aktuelle Asset-Datenbank für ihre Infrastruktur.

Bei Nachrichtendiensten entfallen die Punkte Kosten und Asset-Datenbank, dafür rücken hier die passive Detektion mit großer Sicherheit und in Echtzeit in den Vordergrund.

In den Tabellen 3.7 sowie 3.8 werden die aus den drei Szenarien hergeleiteten Funktionalen und Nicht Funktionalen Anforderungen zusammengeführt (Mehrfachnennungen möglich). Die Priorisierungen der aus den Szenarien hergeleiteten Anforderungen aus den Tabellen 3.1 mit 3.2 (Hochschulrechenzentrum), 3.3 mit 3.4 (Unternehmen) sowie 3.5 mit 3.6 (Strafverfolgungsbehörden und Nachrichtendienste) werden ebenfalls in die neu entstandenen Tabellen übernommen, so dass sich im Anschluss eine Priorisierung der Gesamtanforderungen diskutieren und ableiten lässt. Die Zusammenfassung der Funktionalen und Nicht Funktionalen Anforderungen an das im Rahmen dieser Arbeit entwickelte Flow-Record-Fingerprinting-Tool (FRF-Tool) sowie deren Gesamt-Priorisierung werden abschließend in Tabelle 3.9 sowie in Tabelle 3.10 komprimiert dargestellt.

Tabelle 3.7.: Zusammenfassung Funktionaler Anforderungen in Funktionale Gesamtanforderungen

ID	**Anforderung**	**Szenario 1**	**Szenario 2**	**Szenario 3**
Funktionale Anforderung (FA) 1	Das Tool muss in heterogenen Systemen einsetzbar sein.	FA I (+++)	FA IX (++)	FA XVII (++)
Funktionale Anforderung (FA) 2	Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein.	FA II (++)	FA X (++) FA IX (++)	FA XVIII (+)
Funktionale Anforderung (FA) 3	Die Datenerfassung muss in Grenzpunkten zu autonomen Netzen möglich sein.	FA III (+)		FA XVI (+++)
Funktionale Anforderung (FA) 4	Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein.	FA IV (+++) FA VIII (+)	FA XIII (+++)	
Funktionale Anforderung (FA) 5	Das Tool muss eingesetzte Betriebssysteme sowie Software und deren Versionsstand erkennen.	FA V (+++)	FA XI (+++)	
Funktionale Anforderung (FA) 6	Das Tool muss Assets auch in durch Dritte betreuten Netzen erkennen können.	FA VI (++)		FA XIX (++)
Funktionale Anforderung (FA) 7	Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.	FA VII (+)	FA XIV (+)	
Funktionale Anforderung (FA) 8	Der Einsatz des Tools muss zuverlässig und zeitnah möglich sein, ohne dabei Störungen zu verursachen.	FA VIII (+++)	FA XII (+)	
Funktionale Anforderung (FA) 9	Das Tool muss Rückschlüsse auf Übertragungsinhalte ermöglichen.			FA XV (+++)

Hochschulrechenzentren (Szenario 1), Unternehmen (Szenario 2), Strafverfolgungsbehörden und Nachrichtendienste (Szenario 3)

Tabelle 3.8.: Zusammenfassung Nicht Funktionaler Anforderungen in Nicht Funktionale Gesamtanforderungen

ID	**Anforderung**	**Szenario 1**	**Szenario 2**	**Szenario 3**
Nicht Funktionale Anforderung (NFA) 1	Die Erreichbarkeit, Performance sowie Wartbarkeit bereitgestellter Dienste und Server darf nicht eingeschränkt werden.	NFA I (+++)	NFA X (+++) NFA XI (+++)	
Nicht Funktionale Anforderung (NFA) 2	Die Performance und Leistungsfähigkeit des Netzes darf nicht eingeschränkt werden.	NFA II (+++)	NFA XI (+++) NFA X (+++)	
Nicht Funktionale Anforderung (NFA) 3	Gewonnene Daten müssen für andere Systeme lesbar und auswertbar sein.	NFA III (+)	NFA XII (+) NFA XIII (+)	
Nicht Funktionale Anforderung (NFA) 4	Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden.	NFA IV (+)		
Nicht Funktionale Anforderung (NFA) 5	Vorhandene Infrastruktur und gegebene Möglichkeiten sind zu bevorzugen.	NFA V (+)		
Nicht Funktionale Anforderung (NFA) 6	Das Tool muss moderne Protokolle (IPv6) unterstützen.	NFA VI (++)		NFA XVI (+++)
Nicht Funktionale Anforderung (NFA) 7	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	NFA VII (+++)	NFA XIV (+++)	NFA XIX (+++)
Nicht Funktionale Anforderung (NFA) 8	Bei der Datenerfassung muss der Datenschutz im Sinne des BDSG sowie des BayDSG eingehalten werden.	NFA VIII (+++)	NFA XV (+++)	NFA XX (+++)
Nicht Funktionale Anforderung (NFA) 9	Die Datenerfassung sowie die Analyse muss die Regelungen des StGB einhalten.	NFA IX (+++)		NFA XXI (+++)
Nicht Funktionale Anforderung (NFA) 10	Das Tool muss für Netzteilnehmer transparent sein.		X (+++)	NFA XVII (+++)
Nicht Funktionale Anforderung (NFA) 11	Eine gezielte Auswahl der zu überwachenden und analysierenden Systeme muss vorab möglich sein.			NFA XVIII (+++)

Hochschulrechenzentren (Szenario 1), Unternehmen (Szenario 2), Strafverfolgungsbehörden und Nachrichtendienste (Szenario 3)

3.2.1. Funktionale Gesamtanforderungen

Aus den Funktionalen Anforderungen aus den Tabellen 3.1, 3.3 und 3.5 lassen sich die folgenden Funktionalen Gesamtanforderungen festlegen und priorisieren.

Basierend auf den Funktionalen Anforderungen FA I, FA IX sowie FA XVII geht hervor, dass es für ein Flow-Record-Fingerprinting-Tool (FRF-Tool) notwendig ist, auch mit heterogenen Netzen kompatibel zu sein. Diese Anforderung ist in allen drei Szenarien als wichtig eingestuft worden und wird daher auch als wichtige Gesamtanforderung gesehen.

Funktionale Gesamtanforderung (FA) 1

Das Tool muss in heterogenen Systemen einsetzbar sein. (++)

Aus den Anforderungen von Hochschulrechenzentren, Unternehmen sowie von Strafverfolgungsbehörden und Nachrichtendiensten, welche oft in beziehungsweise mit (geographisch) weitläufigen Netzen arbeiten, lässt sich die Anforderung der Kompatibilität mit eben diesen Netzen ableiten. Hierbei ist festzuhalten, dass diese Anforderung sowohl für Hochschulrechenzentren als auch für Unternehmen auf Grund der oft weitläufigen betreuten Netze von besonderer Wichtigkeit ist. Für Strafverfolgungsbehörden und Nachrichtendienste ist die Einsetzbarkeit in weitläufgen Netzen jedoch nicht zwingend erforderlich, da das Ziel der Überwachung oftmals konkret am Anschlusspunkt beobachtet werden kann. Da die Einsetzbarkeit in weitläufigen Netzen in den Szenarien als wichtig bewertet wurde, folgt auch für die Gesamtanforderung eine hohe Priorisierung der Anforderung.

Funktionale Gesamtanforderung (FA) 2

Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein. (++)

Sowohl für Hochschulrechenzentren als auch für Strafverfolgungsbehörden und Nachrichtendienste ergibt sich die Anforderung, Daten in Grenzpunkten zu autonomen Netzen erfassen und auswerten zu können. Diese Anforderung basiert bei Hochschulrechenzentren darauf, dass durch Institute und angeschlossene Organisationen oft unabhängige Teilnetze betrieben werden. Für Strafverfolgungsbehörden und Nachrichtendienste ergibt sich diese Anforderung, da bei der geheimen Überwachung in der Regel kein Zugriff auf das überwachte Netzsegment besteht. Unternehmen haben diese Anforderung jedoch nicht, da diese das Netz vollständig kontrollieren. Da die Untersuchung von Assets in Netzen Dritter für die Szenarien 1 und 3 von Wichtigkeit ist, wird diese Anforderung als wichtig gewichtet.

Funktionale Gesamtanforderung (FA) 3

Die Datenerfassung muss in Grenzpunkten zu autonomen Netzen möglich sein. (++)

Da sowohl in Hochschul- als auch in Unternehmensnetzen oftmals viele Geräte verbunden sind und zudem hoher Datendurchsatz besteht, ist die Anforderung bei hohem Trafficaufkommen einsetzbar zu sein von sehr hoher Priorität.

Funktionale Gesamtanforderung (FA) 4

Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein. (+++)

Die Anforderung im Netz vorhandene Assets erkennen zu können, ist als sehr wichtig einzustufen, da sowohl für Hochschulrechenzentren als auch für Unternehmen das Wissen über vorhandene Systeme von sehr hoher Bedeutung ist.

Funktionale Gesamtanforderung (FA) 5

> *Das Tool muss eingesetzte Betriebssysteme sowie Software und deren Versionsstand erkennen. (+++)*

Da sowohl Hochschulrechenzentren als auch Strafverfolgungsbehörden und Nachrichtendienste teilweise keine uneingeschränkte Kontrolle über Netzsegmente haben, ist die Anforderung, auch in Netzen Dritter vorhandene Assets zu erkennen, als wichtig einzustufen. Zwar besteht diese Anforderung aus Sicht von Unternehmen nicht, kann jedoch auf Grund der vollständigen Kontrolle von Unternehmen über das eigene Netz nach unten priorisiert werden.

Funktionale Gesamtanforderung (FA) 6

> *Das Tool muss Assets auch in durch Dritte betreuten Netzen erkennen können. (++)*

Die Möglichkeit, das Tool auch in Netzen mit komplexem oder dynamischem Routing einsetzen zu können, wird für Hochschulrechenzentren sowie für Unternehmen benötigt, kann jedoch durch alternative Implementierungen vernachlässigt werden.

Funktionale Gesamtanforderung (FA) 7

> *Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein. (+)*

Eine zeitnahe Detektion ohne Verursachung von Störungen ist sowohl für Hochschulrechenzentren als auch für Unternehmen von Wichtigkeit. Indirekt wird dies durch die Forderung von Strafverfolgungsbehörden und Nachrichtendiensten für die Zielperson transparent zu sein unterstrichen.

Funktionale Gesamtanforderung (FA) 8

> *Der Einsatz des Tools muss zuverlässig und zeitnah möglich sein, ohne dabei Störungen zu verursachen. (++)*

Für den Haupteinsatzzweck aus Sicht von Strafverfolgungsbehörden sowie Nachrichtendiensten wird für Szenario 3 die Möglichkeit Rückschlüsse auf Übertragungsinhalte zu ziehen benötigt. Für Hochschulrechenzentren und Unternehmen wird lediglich die Möglichkeit gefordert, Assets detektieren zu können.

Funktionale Gesamtanforderung (FA) 9

> *Das Tool muss Rückschlüsse auf Übertragungsinhalte ermöglichen. (+)*

Tabelle 3.9.: Funktionale Gesamtanforderungen

ID	**Schlagwort**	**Anforderung**	**Priorität**
FA 1	Heterogenität	Das Tool muss in heterogenen Systemen einsetzbar sein.	++
FA 2	Weitläufigkeit	Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein.	++
FA 3	Grenzerfassung	Die Datenerfassung muss in Grenzpunkten zu autonomen Netzen möglich sein.	++
FA 4	Teilnehmerzahl und Traffic	Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein.	+++
FA 5	Asseterkennung	Das Tool muss eingesetzte Betriebssysteme sowie Software und deren Versionsstand erkennen.	+++
FA 6	Netze Dritter	Das Tool muss Assets auch in durch Dritte betreuten Netzen erkennen können.	++
FA 7	Routing	Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.	+
FA 8	Störungsfreiheit	Der Einsatz des Tools muss zuverlässig und zeitnah möglich sein, ohne dabei Störungen zu verursachen.	++
FA 9	Übertragungsinhalte	Das Tool muss Rückschlüsse auf Übertragungsinhalte ermöglichen.	+

\+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

3.2.2. Nicht Funktionale Gesamtanforderungen

Sowohl für Hochschulrechenzentren als auch für Unternehmen ist es von besonderer Wichtigkeit, dass betriebene Dienste weder beeinträchtigt noch behindert werden. Daher ist diese Anforderung als besonders wichtig einzustufen.

Nicht Funktionale Gesamtanforderung (NFA) 1

> *Die Erreichbarkeit, Performance sowie Wartbarkeit bereitgestellter Dienste und Server darf nicht eingeschränkt werden. (+++)*

Da sowohl für Hochschulrechenzentren als auch für Unternehmen die Leistungsfähigkeit und Performance des bereitgestellten Netzes von besonderer Wichtigkeit sind, ist die Anforderung, das Netz nicht zu belasten, als sehr wichtig einzustufen.

Nicht Funktionale Gesamtanforderung (NFA) 2

> *Die Performance und Leistungsfähigkeit des Netzes darf nicht eingeschränkt werden. (+++)*

Die Anforderung, gewonnene Daten in anderen Systemen weiterverarbeiten zu können, kann für den Erfolg des Flow-Record-Fingerprinting-Tools (FRF-Tools) als nicht notwendig eingestuft werden, da diese Anforderung in allen Szenarien niedrig priorisiert ist.

Nicht Funktionale Gesamtanforderung (NFA) 3

> *Gewonnene Daten müssen für andere Systeme lesbar und auswertbar sein. (+)*

Zwar ist die Anforderung keine zusätzlichen Kosten zu verursachen nachvollziehbar, jedoch wird dies lediglich durch das Hochschulrechenzentrum gefordert und hat dort eine niedrige Priorität. Für Unternehmen ist ein angemessenes Kosten-Nutzen-Verhältnis selbstverständlich. Aus diesem Grund wird die Anforderung insgesamt auch niedrig priorisiert.

Nicht Funktionale Gesamtanforderung (NFA) 4

> *Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden. (+)*

Um Aufwand für die Installation neuer Untersuchungshardware gering zu halten, ist es sinnvoll bestehende Infrastrukturen zu nutzen. Da die Nutzung bestehender Infrastrukur aber nicht zwingend notwendig ist, wird diese Anforderung gering priorisiert.

Nicht Funktionale Gesamtanforderung (NFA) 5

> *Vorhandene Infrastruktur und gegebene Möglichkeiten sind zu bevorzugen. (+)*

Die Anforderung, IPv6 zu unterstützen ist für Strafverfolgungsbehörden und Nachrichtendienste hoch priorisiert. Da das Flow-Record-Fingerprinting-Tool (FRF-Tool) jedoch auch ohne IPv6 einsetzbar ist, kann diese Anforderung als mittel priorisiert werden.

Nicht Funktionale Gesamtanforderung (NFA) 6

> *Das Tool muss moderne Protokolle (IPv6) unterstützen. (++)*

Die Einhaltung geltenden Rechts sowie damit einhergehend insbesondere die Einhaltung der geltenden Datenschutzgesetze ist für das FRF-Tool unabdingbar und stellt ein Kriterium für den Erfolg dar. Aus dieser Anforderung ergeben sich nachfolgend die Datensparsamkeit und Anonymsierung.

Nicht Funktionale Gesamtanforderung (NFA) 7

Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten. (+++)

Der deutsche Datenschutz stellt besondere Anforderungen an die automatisierte Auswertung von personenbezogenen Daten. Im Rahmen der Analyse von Netzkommunikation fallen schützenswerte beziehungsweise besonders schützenswerte Daten an. Entsprechend erfolgt eine sehr hohe Priorisierung.

Nicht Funktionale Gesamtanforderung (NFA) 8

Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes (BDSG) sowie des Bayerischen Datenschutzgesetzes (BayDSG) eingehalten werden. (+++)

Mit der Gesetzeskonformität geht einher, dass die Datenerfassung und -sammlung konform mit geltendem Recht sein muss. Dies gilt explizit für die Untersuchung von Systemen Dritter und geht mit sehr hoher Wichtigkeit einher.

Nicht Funktionale Gesamtanforderung (NFA) 9

Die Datenerfassung sowie die Analyse müssen die Regelungen des StGB einhalten. (+++)

Aus Szenario 3 ergibt sich die Anforderung, für das untersuchte System transparent zu sein. Zusätzlich wird diese Anforderung durch die Nicht Funktionale Gesamtanforderung 1 sowie 2 unterstützt, so dass die Anforderung der Transparenz – obwohl diese lediglich durch Szenario 3 gefordert wird – als sehr wichtig für das Flow-Record-Fingerprinting-Tool (FRF-Tool) eingestuft wird.

Nicht Funktionale Gesamtanforderung (NFA) 10

Das Tool muss für Netzteilnehmer transparent sein. (+++)

Da die Anforderung, dass eine gezielte Analyse der unterschiedlichen Systeme möglich ist, lediglich durch Szenario 3 gefordert wird, jedoch für die weiteren Szenarien nicht von Bedeutung ist, wird die Gesamtanforderung als nicht für das FRF-Tool zwingend notwendig eingestuft.

Nicht Funktionale Gesamtanforderung (NFA) 11

Eine gezielte Auswahl der zu überwachenden und analysierenden Systeme muss vorab möglich sein. (+)

Tabelle 3.10.: Nicht Funktionale Gesamtanforderungen

ID	**Schlagwort**	**Anforderung**	**Priorität**
NFA 1	Verfügbarkeit	Die Erreichbarkeit, Performance sowie Wartbarkeit bereitgestellter Dienste und Server darf nicht eingeschränkt werden.	+++
NFA 2	Performance	Die Performance und Leistungsfähigkeit des Netzes darf nicht eingeschränkt werden.	+++
NFA 3	Auswertbarkeit	Gewonnene Daten müssen für andere Systeme lesbar und auswertbar sein.	+
NFA 4	Kosten	Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden.	+
NFA 5	Vorhandene Infrastruktur	Vorhandene Infrastruktur und gegebene Möglichkeiten sind zu bevorzugen.	+
NFA 6	IPv6	Das Tool muss moderne Protokolle (IPv6) unterstützen.	++
NFA 7	Anonymisierung	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	+++
NFA 8	Datenschutz	Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes (BDSG) sowie des Bayerischen Datenschutzgesetzes (BayDSG) eingehalten werden.	+++
NFA 9	Gesetzeskonformität	Die Datenerfassung sowie die Analyse müssen die Regelungen des StGB einhalten.	+++
NFA 10	Transparenz	Das Tool muss für Netzteilnehmer transparent sein.	+++
NFA 11	Selektion	Eine gezielte Auswahl der zu überwachenden und analysierenden Systeme muss vorab möglich sein.	+

+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

4. Themenverwandte Arbeiten

Um eingesetzte Software, genutzte Betriebssysteme oder Übertragungsinhalte in Rechnernetzen zu erkennen, stehen bereits verschiedene Methoden zur Verfügung. Nachfolgend werden diese Methoden kurz vorgestellt und auf ihre Eignung zur Erfüllung der Gesamtanforderungen und Ziele dieser Arbeit untersucht. Generell lassen sich die bisher bekannten Methoden in zwei Kategorien einteilen, nämlich die aktiven und die passiven Erkennungsmethoden. Ein hybrides Herangehen ist ebenfalls möglich. An Hand verschiedener aktueller Arbeiten werden die auf den unterschiedlichen Erkennungsmethoden basierende Verfahren kurz vorgestellt und diskutiert.

4.1. Aktive Detektionsverfahren

Bei den aktiven Detektionsverfahren gibt es verschiedene Vertreter und Hilfsmittel. Als Beispiele lassen sich das Tool `Network Mapper (Nmap)` oder als hierauf aufbauendes Verfahrenskonzept `Dr. Portscan` aufführen. Neben diesen beiden existieren noch weitere auf `Nmap` oder ähnlicher Software aufsetzende Toolsets.

Der entscheidende Nachteil von aktiven Verfahren ist, dass die untersuchten Systeme in Mitleidenschaft gezogen werden können. So können bei einem Scanversuch durch zu hohe Raten an (SYN-)Anfragen zentrale Elemente wie Router im Betrieb gestört oder unnötige Logdateien erstellt werden. Die Einführung der 6. Version des Internetprotokolls (IPv6), durch die der verfügbare Adressraum von 4.294.967.296 Adressen in IPv4 auf über $3,4 * 10^{38}$ erhöht wurde, stellt ein weiteres Problem für aktive Scanverfahren dar. Die so entstandene Erhöhung der Anzahl möglicher Adressen in einem Rechnernetz steigert die Komplexität eines vollständigen Netzscans derart, dass eine Untersuchung eines vollständigen Netzes durch einen Analyserechner nicht mehr effektiv möglich ist.

4.1.1. Xprobe

In seiner Arbeit „A remote active OS fingerprinting tool using ICMP“ stellt Ofir Arkin die Möglichkeit vor, mittels ICMP-Anfragen auf das eingesetzte Betriebssystem zu schließen.

Hierfür entwickelte Ofir Arkin das Tool Xprobe, welches die Antworten des TCP/IP-Stacks des zu untersuchenden Systems analysiert. Zur Untersuchung werden durch Xprobe verschiedene ICMP-Anfragen versendet. Hierbei baut die Untersuchung darauf auf, dass einige Betriebssystemstacks sich in der Anfragebeantwortung durch die genutzten Feldwerte unterscheiden. Somit ist es Xprobe möglich, durch eine geringe Anzahl an Anfragen – sofern nicht durch die Firewall des Zielsystems blockiert – das genutzte System zu detektieren [Ark02].

Aufbauend auf der Weiterentwicklung von Xprobe untersuchten Ofir Arkin und Fyodor Yarochkin die Möglichkeiten, die Xprobe2 zur Betriebssystemanalyse bietet. Hierbei stellten sie in ihrer Arbeit „A “Fuzzy” Approach to Remote Active Operating System Fingerprinting“ die Möglichkeit der Erweiterung der Anfragen durch Xprobe mittels der neu

entstandenen API um systemspezifische Anfragen vor. Des Weiteren nutzten sie hierbei verschiedene Anfragen, um einen Score für das vermutlich genutzte Betriebssystem zu erstellen. Durch diese Weiterentwicklung wurde Xprobe flexibler einsetzbar und an neue Umstände anpassbar [AY02].

Das von Xprobe genutzte Verfahren widerspricht jedoch verschiedenen Anforderungen dieser Arbeit. So ist es weder möglich die eingesetzte Software noch die übertragenen Daten zu detektieren. In größeren Netzen wie beispielsweise dem Münchner Wissenschaftsnetz (MWN) ist es zudem notwendig, jedes System einzeln anzusprechen und somit sehr viele Anfragen zu stellen, um das gesamte Netz untersuchen zu können. Der so entstehende Traffic kann das Netz in seiner Qualität beeinträchtigen und ist mit einer hohen Laufzeit verbunden.

Nachfolgend wird Xprobe mit den Gesamtanforderungen abgeglichen.

FA 1 Heterogenität
Die Anforderung in heterogenen Netzen, also in Netzen in denen verschiedene Systeme (sowohl Hardware als auch Software) angeschlossen sind, betrieben werden zu können, wird durch Xprobe erfüllt.

FA 2 Weitläufigkeit
Die Anforderung in weitläufigen Netzen nutzbar zu sein, wird von Xprobe erfüllt. Routing oder zwischengelagerte Router beeinflussen die ICMP-Antwort des angefragten Rechners nicht, so dass auch über weitläufige Netze noch die ursprünglichen Antworten erfasst werden können.

FA 3 Grenzerfassung
Da Xprobe ein rein aktiver Netzscanner ist, ist eine passive Aufzeichnung nicht möglich, so dass diese Anforderung nicht erfüllt wird.

FA 4 Teilnehmerzahl und Traffic
Die Anforderung in einer Umgebung mit vielen Geräten und hohem Trafficaufkommen zu funktionieren, wird nur teilweise erfüllt. Da es jedoch erforderlich ist, alle Hosts einzeln zu scannen, nehmen mit zunehmender Teilnehmerzahl Komplexität und Laufzeit zu.

FA 5 Asseterkennung
Die Erkennung von Assets ist mit einem Tool wie Xprobe teilweise möglich, sofern diese Assets auf die durch `Nmap` versendeten Anfragen antworten. Im Fall einer Antwort ist die Detektion von Betriebssystem sowie Betriebssystemversion möglich. Eine Detektion von installierter Software sowie betriebenen Diensten ist nicht möglich.

FA 6 Netze Dritter
Sofern Xprobe Scans in Netzen Dritter durchführen kann und hierbei nicht durch Firewalls blockiert wird, ist diese Anforderung erfüllt.

FA 7 Routing
Die Anforderung mit dynamischem Routing kompatibel zu sein, wird von Xprobe erfüllt, solange der Xprobe-Server alle Clients erreichen kann.

FA 8 Störungsfreiheit
Die Anforderung, Teilnehmer im Netz durch die Analyse nicht zu stören, wird durch Xprobe nur teilweise erfüllt, da der Scan unter Umständen unnötige Logdateien provozieren kann.

FA 9 Übertragungsinhalte
Da Xprobe ein aktiver Scan ist, ist es nicht möglich, Übertragungsinhalte zu erkennen. Somit wird die Anforderung nicht erfüllt.

NFA 1 Verfügbarkeit
Da Xprobe nur eine geringe Anzahl an Anfragen versendet, welche das Zielsystem in der Regel nicht belasten, wird diese Anforderung erfüllt.

NFA 2 Performance
Da Xprobe nur eine geringe Anzahl an Anfragen versendet und diese das Netz kaum belasten, wird diese Anforderung erfüllt.

NFA 3 Auswertbarkeit
Die Anforderung wird erfüllt, da Xprobe auswertbare Ausgaben liefert.

NFA 4 Kosten
Die Anforderung keine Kosten zu verursachen wird – sofern keine zusätzliche Hardware benötigt wird – erfüllt.

NFA 5 Vorhandene Infrastruktur
Die Anforderung, die vorhandene Infrastruktur zu nutzen, wird durch Xprobe so lange erfüllt, wie keine neue Hardware für Scan oder Netzperformence benötigt wird.

NFA 6 IPv6
Xprobe unterstützt zwar IPv6, jedoch stellt die Zahl der möglichen Hosts in einem IPv6-Netz eine derartige Komplexität dar, dass diese Kompatibilität nur bedingt gegeben und somit die Anforderung nur teilweise erfüllt ist.

NFA 7 Anonymisierung
Die Einhaltung gesetzlicher Anforderungen wie Anonymisierung ist keines der Funktionsmerkmale von Xprobe, da dieses für die Anwendung im eigenen Netz entwickelt wurde. Somit wird die Anforderung nicht erfüllt.

NFA 8 Datenschutz
Auch die Einhaltung gesetzlicher Anforderungen wie Datenschutz ist keines der Funktionsmerkmale von Xprobe. Somit wird die Anforderung nicht erfüllt.

NFA 9 Gesetzeskonformität
Da Xprobe – wie bereits erwähnt – für den Einsatz im eigenen Netz entwickelt wurde, stellt die Gesetzeskonformität keines der Funktionsmerkmale von Xprobe dar. Somit wird die Anforderung nicht erfüllt.

NFA 10 Transparenz
Die Anforderung, für das gescannte System transparent zu sein, wird auf Grund der aktiven Anfragen von Xprobe nicht erfüllt.

NFA 11 Selektion
Xprobe erlaubt es, gezielt Hosts auszuwählen und erfüllt somit die Nicht Funktionale Anforderung 11.

Tabelle 4.1.: Anforderungscheck Xprobe

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	nicht erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	teilweise erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	teilweise erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	erfüllt
NFA 6	IPv6	teilweise erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	nicht erfüllt
NFA 10	Transparenz	nicht erfüllt
NFA 11	Selektion	erfüllt

4.1.2. nmap

Das weit verbreitete Tool `Nmap`, welches für Service Discovery und Systemanalyse genutzt werden kann, nutzt unter anderem aktives Probing, indem gezielt präparierte Pakete an ein zu identifizierendes System gesendet werden. Die auf die gesendeten Pakete erhaltenen Antworten werden erfasst und mit einer Signaturdatenbank verglichen. Zur Steigerung der Effizienz und Vermeidung von überflüssigen Anfragen wird in der Regel zuerst ein Host Discovery zur Feststellung der mit dem Netz verbundenen Rechner und anschließend ein Portscan auf den identifizierten Rechnern durchgeführt.

Ein besonderer Vorteil von `Nmap` ist, dass es sich hierbei um einen Open-Source Sicherheitsscanner handelt, der für die Detektion von sich im Netz befindlichen Systemen und dort betriebener Dienste konzipiert wurde. Hierfür bietet `Nmap` die Möglichkeiten Hosts in einem Netz zu erkennen, Ports zu scannen, Software und Versionen sowie Betriebssysteme zu detektieren. In den Standardeinstellungen untersucht `Nmap` alle Ports eines Hosts, indem es TCP SYN-Pakete, den sogenannten TCP-SYN-Scan, an den untersuchten Port sendet. Falls dieser Port geöffnet ist, wird mit einem TCP SYN+ACK geantwortet, andernfalls mit einem TCP RST. Durch diese Untersuchungstechnik lässt sich schnell herausfinden, welche Ports an einem Host geöffnet sind.

Um weitere Informationen über den untersuchten Host zu generieren, versucht `Nmap`, den genutzten Service sowie dessen Version durch Senden weiterer Prüfpakete herauszufinden. So wird hierbei unter anderem eine Verbindung aufgenommen und auf die Antwort des Dienstes für eine gewisse Zeit gewartet, da viele Dienste sich selbst mit einer Willkommensnachricht identifizieren. Die empfangenen Daten werden anschließend mit einer Signatur-

datenbank verglichen. Wurden keine Daten empfangen, werden weitere Prüfbefehle an den Dienst gesendet, um Antworten zu provozieren. Ein derartiger Befehl ist zum Beispiel ‘help \n \r‘ [TThCcC09], wodurch in der Regel die Hilfeseite des Dienstes ausgegeben wird.

Größtes Problem bei Netzanalysen mittels Nmap stellen die vielen Anfragen mit dem hierbei entstehenden Traffic dar. Durch den Scan eines einzelnen Rechners werden bis zu 65.535 Ports untersucht, wobei für jeden Port 1 bis 30 Pakete gesendet werden können. Somit widerspricht ein blinder Scan des vollständigen Netzes den Kriterien der Durchsichtigkeit und Nicht-Störung des Netzes. Ferner können derartige Scans unnötige Logdateien oder Abwehrmechanismen auf dem Zielhost verursachen.

FA 1 Heterogenität
Die Anforderung in heterogenen Netzen, also in Netzen in denen verschiedene Systeme (sowohl Hardware als auch Software) angeschlossen sind, betrieben werden zu können, wird von `Nmap` erfüllt.

FA 2 Weitläufigkeit
Die Anforderung in weitläufigen Netzen nutzbar zu sein, wird von `Nmap` teilweise erfüllt. Es ist zwar möglich in weitläufigen Netzen einzelne Hosts zu scannen, doch nehmen die Hostanzahl oder die Verzögerung und Distanz zu, so ist `Nmap` nur noch langsam nutzbar.

FA 3 Grenzerfassung
Da `Nmap` ein rein aktiver Netzscanner ist, ist eine passive Aufzeichnung nicht möglich, so dass diese Anforderung nicht erfüllt wird.

FA 4 Teilnehmerzahl und Traffic
Die Anforderung, in Umgebungen mit vielen Geräten und hohem Trafficaufkommen zu funktionieren, wird nur teilweise erfüllt. Da `Nmap` aktiv scannt, ist der durch Clients anfallende Traffic für `Nmap` nicht von Bedeutung, jedoch wird die Dauer eines vollständigen Scans mit jeder Teilnehmervermehrung erhöht, so dass der Einsatz von `Nmap` für vollständige Scans irgendwann nicht mehr zielführend ist.

FA 5 Asseterkennung
Die Erkennung von Assets ist mit einem Tool wie `Nmap` möglich, sofern diese auf die durch `Nmap` versendeten Anfragen antworten. Im Fall einer Antwort ist die Detektion von Betriebssystem sowie Betriebssystemversion möglich.

FA 6 Netze Dritter
Sofern `Nmap` Scans in Netzen Dritter durchführen kann und hierbei nicht durch Firewalls blockiert wird, ist diese Anforderung erfüllt.

FA 7 Routing
Die Anforderung, mit dynamischem Routing kompatibel zu sein, wird von `Nmap` erfüllt, solange immer eine bidirektionale Verbindung zwischen `Nmap`-Server und Host besteht.

FA 8 Störungsfreiheit
Die Anforderung, Teilnehmer im Netz durch die Analyse nicht zu stören, wird durch `Nmap` nicht erfüllt, da der Scan unter Umständen unnötige Logdateien provozieren beziehungsweise Netzgeräte durch die Anzahl der Anfragen eines breiten Scans überlasten kann.

FA 9 Übertragungsinhalte
Da `Nmap` ein aktives Scanverfahren ist, ist es nicht möglich Übertragungsinhalte zu erkennen. Somit wird die Anforderung nicht erfüllt.

NFA 1 Verfügbarkeit
Durch den aktiven Scan und eine potentielle Belastung des Zielsystems können Dienstperformance und Verfügbarkeit eingeschränkt werden. Somit ist die Anforderung nur teilweise erfüllt.

NFA 2 Performance
`Nmap` kann Traffic verursachen und somit die Performance des Netzes negativ beeinflussen. Somit wird die Anforderung nur teilweise erfüllt.

NFA 3 Auswertbarkeit
Die Anforderung wird erfüllt, da `Nmap` Ausgaben liefert, die weiter ausgewertet werden können.

NFA 4 Kosten
Die Anforderung keine Kosten zu verursachen wird – sofern keine Hardware benötigt wird – erfüllt.

NFA 5 Vorhandene Infrastruktur
Die Anforderung, die vorhandene Infrastruktur zu nutzen, wird durch `Nmap` teilweise erfüllt, da auf Grund der Komplexität und Dauer der einzelnen Hostscans gegebenenfalls neue Scan-Server benötigt werden, um weitläufige Scans durchführen zu können.

NFA 6 IPv6
`Nmap` unterstützt zwar IPv6, jedoch stellt die Zahl der möglichen Hosts in einem IPv6-Netz eine derartige Komplexität dar, dass diese Kompatibilität nur bedingt gegeben und somit die Anforderung nur teilweise erfüllt ist.

NFA 7 Anonymisierung
Die Einhaltung gesetzlicher Anforderungen, wie unter anderem die Anonymisierung der erfassten Daten, ist keines der Funktionsmerkmale von `Nmap`, da dieses Tool für die Anwendung im eigenen Netz entwickelt wurde. Somit wird die Anforderung nicht erfüllt.

NFA 8 Datenschutz
Die Einhaltung des gesetzlichen Datenschutzes ist ebenfalls keines der Funktionsmerkmale von `Nmap`, da dieses Tool für die Anwendung im eigenen Netz entwickelt wurde. Somit wird die Anforderung nicht erfüllt.

NFA 9 Gesetzeskonformität
Auch die Einhaltung gesetzlicher Rahmenbedingungen, wie beispielsweise des StGB, ist keines der Funktionsmerkmale von `Nmap`. Somit wird die Anforderung nicht erfüllt.

NFA 10 Transparenz
Die Anforderung, für das gescannte System transparent zu sein, wird auf Grund des aktiven Scans von `Nmap` nicht erfüllt.

NFA 11 Selektion
`Nmap` erlaubt es, gezielt Hosts auszuwählen und erfüllt somit die Nicht Funktionale Anforderung 11.

Tabelle 4.2.: Anforderungscheck `Nmap`

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	teilweise erfüllt
FA 3	Grenzerfassung	nicht erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	nicht erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	teilweise erfüllt
NFA 2	Performance	teilweise erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	teilweise erfüllt
NFA 6	IPv6	teilweise erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	nicht erfüllt
NFA 10	Transparenz	nicht erfüllt
NFA 11	Selektion	erfüllt

4.1.3. Dr. Portscan

Beim durch das LRZ entwickelten `Dr. Portscan` handelt es sich nicht um ein aktives Netzanalysepaket im eigentlichen Sinn, sondern um ein Portscan-Reportingwerkzeug. Hierbei ist der Hauptzweck von `Dr. Portscan` nicht das aktive Untersuchen eines Netzes, sondern die Zusammenführung, Bereitstellung und Auswertung der generierten Daten verschiedener Netzscanner und Analysewerkzeuge.

Historisch geht `Dr. Portscan` auf eine Sammlung von Perl- und Shellskripten zurück, die zur Gegenüberstellung von Nmap-Resultaten genutzt wurden. Die Nmap Scans werden hierbei von verschiedenen Systemen aus erzeugt. Die Unübersichtlichkeit der generierten Daten sowie die Unterschiedlichkeit der Ausgabeformate verschiedener Analysetools machen die Entwicklung eines Werkzeugs für die automatisierte Portscan-Auswertung in komplexen Netzinfrastrukturen notwendig. Abbildung 4.1 stellt dar, wie vor der Einrichtung von `Dr. Portscan` das Netz und die vorhandenen Assets untersucht wurden.

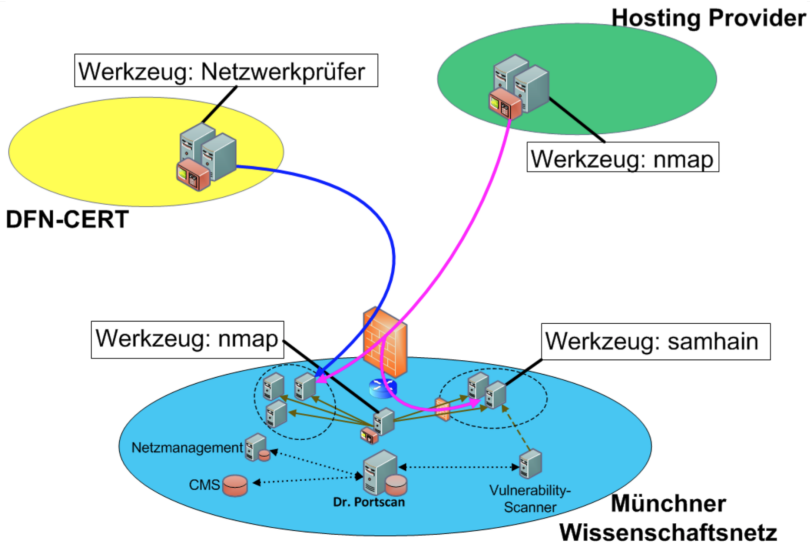


Abbildung 4.1.: Ausgangssituation vor der Einführung von Dr. Portscan [vMH13]

Demgegenüber verdeutlicht Abbildung 4.2 die nach Einführung von `Dr. Portscan` entstandene Situation, in der die Anforderung, Ergebnisse verschiedener Tools mit Messpunkten innerhalb und außerhalb des Netzes zusammenzuführen und auszuwerten, umgesetzt wurde. In dieser Abbildung ist auch dargestellt, wie die verschiedenen Scan-Tools (1) ihre Daten an die zuständigen Input-Agenten (2), welche das Datenformat vereinheitlichen, übertragen. Von dort aus werden die vereinheitlichten Daten an den Delta-Reporter (3) übermittelt. An dieser Stelle (3) wird der Unterschied zwischen Ist-Stand und letztem bekannten Stand erfasst und in eine Datenbank (4) gespeichert. Im Anschluss können durch verschiedene Output-Agenten (5) Aktionen wie beispielsweise das Ausführen von Skripten, das Erstellen von Analyseseiten oder die Information an die zuständigen Administratoren veranlasst werden.

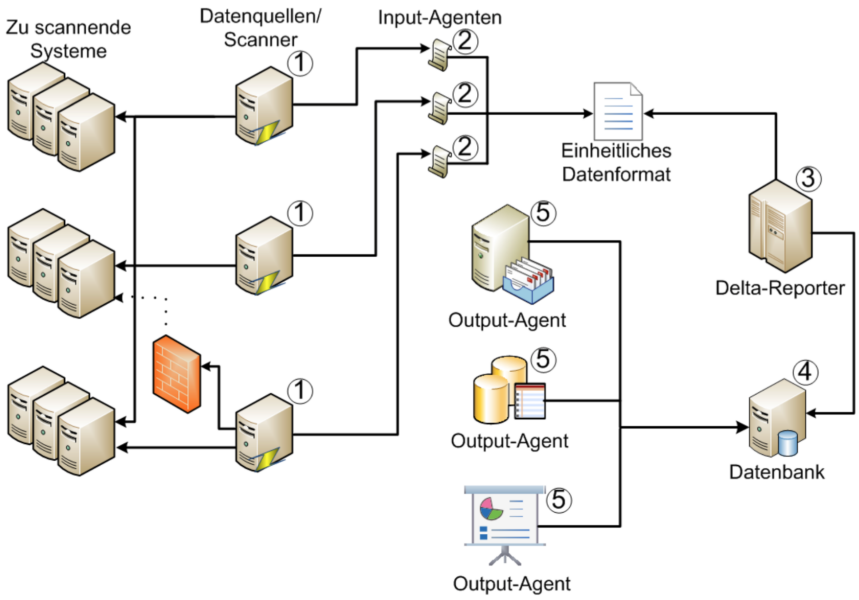


Abbildung 4.2.: Netzanalyse unter Nutzung von `Dr. Portscan` [vMH13]

Zwar ermöglicht der Ansatz von `Dr. Portscan` eine Verteilung der Belastung des Netzes, indem bekannte Dienste gespeichert werden und die Netzanalyse kostensparend unternommen wird, jedoch sieht der Ansatz hinter `Dr. Portscan` eine regelmäßige aktive Untersuchung des Netzes vor. Beim durch das LRZ entwickelten und von Felix von Eye vorgestellten `Dr. Portscan` wird des Weiteren versucht, die Belastung des Netzes sowie des Hosts durch Eingrenzen des zu scannenden Port-Ranges gering zu halten. Dennoch basiert der Ansatz primär auf einer aktiven Netzanalyse, die regelmäßig durchgeführt wird, und hierbei mit zusätzlichem Traffic und Beeinflussung der untersuchten Systeme einhergeht. Somit wird die Anforderung der Transparenz durch dieses Tool nicht erfüllt.

FA 1 Heterogenität
Die Anforderung in heterogenen Systemen einsetzbar zu sein, wird durch `Dr. Portscan` erfüllt, da es sich bei `Dr. Portscan` eher um ein Zusammenfassungs-, Übersichts- und Kontrollwerkzeug als um ein eigentliches Analysetool handelt.

FA 2 Weitläufigkeit
Da `Dr. Portscan` speziell für die Anforderungen des MWN entwickelt worden ist, wurde die Anforderung mit weitläufigen Netzen kompatibel zu sein, bereits zu Beginn beachtet und implementiert.

FA 3 Grenzerfassung
Die Anforderung an Grenzpunkten zu autonomen Netzen Daten zu erfassen, liegt nicht in den durch `Dr. Portscan` umgesetzten Zielen und ist somit nicht erfüllt.

FA 4 Teilnehmerzahl und Traffic
Da `Dr. Portscan` speziell für die Anforderungen des MWN zugeschnitten wurde, ist es mit einer hohen Systemzahl sowie hohem Traffic einsetzbar. Als Einschränkung muss festgehalten werden, dass `Dr. Portscan` für den Einsatz in Rechenzentren, aber nicht für einen Einsatz im Bereich der Clients implementiert wurde. Aus diesem Grund lässt sich diese Anforderung als teilweise erfüllt einordnen.

FA 5 Asseterkennung
Ziel von `Dr. Portscan` ist das Reporting über Portscans und damit die Beantwortung der Frage, welche Ports nach außen hin geöffnet sind. Die Erfassung von Assets liegt nicht im Funktionsumfang von Dr. Portscan, so dass die Anforderung nicht erfüllt wird.

FA 6 Netze Dritter
Die Anforderung in Netzen Dritter einsetzbar zu sein, wird durch `Dr. Portscan` teilweise umgesetzt. Dies ist durch die Erweiterbarkeit von `Dr. Portscan` um weitere Tools erfüllt.

FA 7 Routing
Die Anforderung ist erfüllt, da `Dr. Portscan` entsprechend auf `Nmap` und weiteren Tools aufbaut.

FA 8 Störungsfreiheit
Die Anforderung der Störungsfreiheit ist wesentlicher Bestandteil von Dr. Portscan, so dass diese Anforderung erfüllt wird.

FA 9 Übertragungsinhalte
Ziel von `Dr. Portscan` ist das Reporting über Portscans, also welche Ports nach außen hin geöffnet sind. Die Erfassung und Analyse von Übertragungsinhalten liegt nicht im Funktionsumfang von Dr. Portscan, so dass die Anforderung nicht erfüllt wird.

NFA 1 Verfügbarkeit
Da `Dr. Portscan` für den Einsatz in Rechenzentren entwickelt wurde, ist eine Einschränkung der Verfügbarkeit als unwahrscheinlich zu betrachten.

NFA 2 Performance
Da `Dr. Portscan` für den Einsatz in Rechenzentren entwickelt wurde, ist eine Einschränkung der Netzperformance als unwahrscheinlich zu betrachten.

NFA 3 Auswertbarkeit
Die Anforderung ist als eines der Hauptziele von `Dr. Portscan` erfüllt.

NFA 4 Kosten
Die Anforderung ist teilweise erfüllt, da zwar keine Lizenzkosten, dafür aber Kosten für weitere Server anfallen.

NFA 5 Vorhandene Infrastruktur
Die Anforderung ist nicht erfüllt, da für `Dr. Portscan` spezielle Server innerhalb und außerhalb des Netzes benötigt werden.

NFA 6 IPv6
Auf Grund der in `Dr. Portscan` genutzten Tools (u. a. `Nmap`) ist die Anforderung erfüllt.

NFA 7 Anonymisierung
Da `Dr. Portscan` für den Betrieb in Rechenzentren entwickelt worden ist, wurde die Anonymisierung der Daten hier nicht beachtet.

NFA 8 Datenschutz
Da `Dr. Portscan` für den internen Einsatz entwickelt wurde, wurde nicht auf die Gesetzeskonformität geachtet. Ferner gilt hier die Bewertung von `Nmap`. Somit ist die Anforderung nicht erfüllt.

NFA 9 Gesetzeskonformität
Da `Dr. Portscan` für den internen Einsatz entwickelt wurde, wurde nicht auf die Gesetzeskonformität geachtet. Ferner gilt hier die Bewertung von `Nmap`. Somit ist die Anforderung nicht erfüllt.

NFA 10 Transparenz
Da `Dr. Portscan` aktive Technologien nutzt, ist die Anforderung der Transparenz nicht erfüllt.

NFA 11 Selektion
Die Anforderung der Selektion einzelner Systeme ist kein Bestandteil von Dr. Portscan, so dass diese Anforderung nicht erfüllt ist.

Tabelle 4.3.: Anforderungscheck Dr. Portscan

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	nicht erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	nicht erfüllt
FA 6	Netze Dritter	teilweise erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	teilweise erfüllt
NFA 5	Vorhandene Infrastruktur	nicht erfüllt
NFA 6	IPv6	erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	nicht erfüllt
NFA 10	Transparenz	nicht erfüllt
NFA 11	Selektion	nicht erfüllt

4.1.4. Bewertung aktiver Verfahren

Beispielhaft wurden verschiedene aktive Untersuchungsverfahren beziehungsweise Analysetools vorgestellt, welche grundlegend darauf aufbauen, dass durch den Analyseserver spezielle Pakete an das Zielsystem gesendet werden, während die Antworten hierauf mit einer bestehenden Datenbank abgeglichen werden. Als zentrale Schwachstelle dieser Verfahren konnten sowohl die Laufzeitverlängerung bei Zunahme der Teilnehmer als auch die Störung der bestehenden Infrastruktur durch Logdateien und Traffic aufgezeigt werden. Als weitere Schwachstelle lässt sich festhalten, dass nur Hosts beziehungsweise Dienste, welche auf entsprechende Anfragen antworten, erkannt werden können.

Durch Abbildung 4.3 wird verdeutlicht, dass die in Kapitel 4.1 vorgestellten Verfahren und Tools die im Rahmen dieser Thesis erarbeiteten Anforderungen auch kombiniert nicht oder nur teilweise erfüllen. Hierfür werden in Abbildung 4.3 die Resultate der Analyse übereinandergelegt.

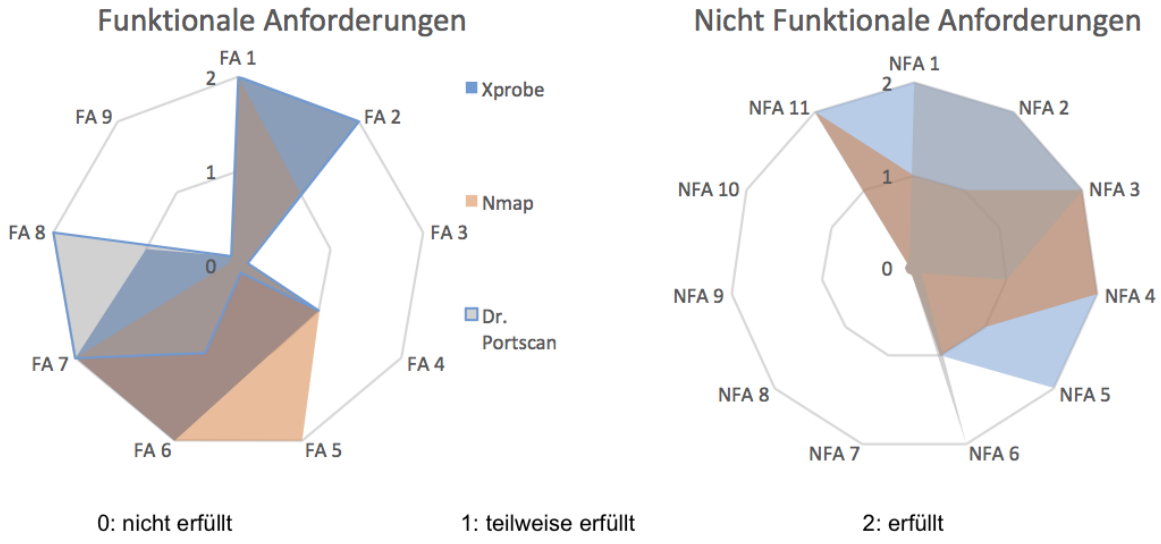


Abbildung 4.3.: Visualisierung der Anforderungserfüllung durch aktive Verfahren

4.2. Passive Detektionsverfahren

Im Gegensatz zu aktiven Verfahren, bei denen das Zielsystem aktiv gescannt wird, basieren passive Erkennungsmethoden auf der Analyse der übertragenen Daten. Für das untersuchte System sind passive Detektionsverfahren transparent, werden also von diesem nicht bemerkt.

4.2.1. PRADS

`Passive Real-time Asset Detection System (PRADS)` ist ein Programm, das den Netzverkehr aufzeichnet und Informationen über Hosts und die auf diesen betriebenen Diensten sammelt. Auf Grund der passiven Analyse können hierbei lediglich aktiv am Netz teilnehmende Systeme erkannt werden. `PRADS` kann sowohl in Echtzeit den anfallenden Datenverkehr analysieren als auch in einem speziellen Format aufgezeichnete Daten auswerten. Zur Erkennung und Analyse werden unter anderem folgende Techniken genutzt:

- OS-Fingerprinting, sowohl SYN als auch SYN+ACK (IP/TCP)
- TCP-Diensterkennung durch Fingerprinting
- TCP-Erkennung von Hosts (SYN und SYN+ACK)
- UDP-Erkennung von Hosts
- UDP OS Fingerprinting

Wird `PRADS` mit dem Befehl prads -i eth0 -l prads.log gestartet, so lauscht das Programm auf den Netzwerkadapter eth0 und schreibt die Ergebnisse in die Logdatei prads.log. Im Anschluss können diese Ergebnisse weiter ausgewertet werden, um zum Beispiel eine Asset-Datenbank zu erstellen [Ber14]. Besonders hervorzuheben ist, dass `PRADS` mit IPv6 kompatibel ist [Ubu].

FA 1 Heterogenität
Als Assetdetektionssystem ist `PRADS` dafür entwickelt, in heterogenen Systemen eingesetzt zu werden; somit wird diese Anforderung erfüllt.

FA 2 Weitläufigkeit
Da `PRADS` auf einem einzelnen Gateway beziehungsweise Überwachungsserver, an welchen der Traffic gespiegelt werden muss, eingesetzt wird, erhöht sich die Komplexität mit zunehmender Weitläufigkeit des Netzes. Da der Traffic hier (meist kostspielig) zum `PRADS`-Server geleitet werden muss und dieser ein Bottle-Neck darstellt, ist `PRADS` mit einer einzelnen Instanz nur bedingt für den Einsatz in weitläufigen Netzen geeignet. Durch Betreiben weiterer Instanzen näher an den zu überwachenden Netzsegmenten lässt sich diesem Problem entgegenwirken, jedoch entstehen hierbei Kosten für die entsprechende Hardware. Aus diesen Gründen erfüllt `PRADS` die Anforderung nur teilweise.

FA 3 Grenzerfassung
Da `PRADS` Assets durch den analysierten Netzverkehr erkennen kann – ohne auf die betroffenen Assets zuzugreifen – ist es generell möglich, `PRADS` an Grenzpunkten zu autonomen Netzen zu einzusetzen. Somit wird diese Anforderung erfüllt.

FA 4 Teilnehmerzahl und Traffic
Da der gesamte zu analysierende Datenverkehr den PRADS-Server durchlaufen beziehungsweise zu diesem hingespiegelt werden muss, erhöhen sich die Kosten mit zunehmender Teilnehmerzahl und steigendem Traffic. Bei zu starkem Trafficanstieg stellt der PRADS-Server eine Engstelle dar, so dass entweder das Netz gebremst oder nicht der vollständige Traffic untersucht wird. Somit ist diese Anforderung teilweise erfüllt.

FA 5 Asseterkennung
Da die Erkennung von Betriebssystemen die Hauptaufgabe von PRADS ist und damit installierte Software sowie betriebene Dienste bisher nicht detektiert werden, ist diese Anforderung teilweise erfüllt.

FA 6 Netze Dritter
Da es sich um eine passive Detektion des anfallenden Traffics handelt, ist die Anforderung der Kompatibilität mit Netzen Dritter erfüllt.

FA 7 Routing
Sofern ein geeigneter Erfassungspunkt für Daten gewählt wird, ist die Anforderung, mit komplexem und dynamischem Routing kompatibel zu sein, erfüllt.

FA 8 Störungsfreiheit
Der Einsatz von PRADS , wie in der Anforderung FA 8 verlangt, ist ohne Störung anderer möglich, sofern verschiedene Bedingungen eingehalten werden. Hierzu zählt insbesondere die Vermeidung einer kostspieligen und weitläufigen Übertragung des Datenverkehrs vom Erfassungsort hin zu einem Analyseserver. Somit ist diese Anforderung erfüllt.

FA 9 Übertragungsinhalte
Da die Erkennung von Übertragungsinhalten kein Bestandteil von PRADS ist, wird diese Anforderung nicht erfüllt.

NFA 1 Verfügbarkeit
Da PRADS lediglich eine passive Analyse des erfassbaren Traffics durchführt, wird die Verfügbarkeit der untersuchten Systeme nicht beeinträchtigt.

NFA 2 Performance
Sofern für PRADS keine Spiegelung des Traffics durch einen Mirrorport sowie kein Transport über das Netz hin zu einem Analyseserver notwendig ist, findet keine negative Beeinträchtigung der Netzperformance statt. Da jedoch abhängig von der jeweiligen Konfiguration eine negative Beeinflussung möglich ist, wird diese Anforderung als teilweise erfüllt gewertet.

NFA 3 Auswertbarkeit
Die Forderung nach einem auswertbaren Ausgabeformat wird durch PRADS erfüllt.

NFA 4 Kosten
Sofern die bestehenden Router zur Erfassung der Daten mittels PRADS genutzt werden und keine zusätzliche Hardware benötigt wird, ist die Anforderung, zusätzliche Kosten zu vermeiden, erfüllt, da PRADS kostenfrei über die Systemrepositories von Ubuntu installiert werden kann.

NFA 5 Vorhandene Infrastruktur
PRADS benötigt für die Erfassung der Daten und deren Analyse einen Server, auf dem PRADS installiert ist. Hierfür lassen sich bestehende Gateways nutzen, sofern deren Performance ausreichend ist. Soll die Datenerfassung zentral erfolgen, so werden Spiegelports und gegebenenfalls mehr Bandbreite benötigt. Damit ist diese Anforderung teilweise erfüllt.

NFA 6 IPv6
Nach Angabe der Entwickler wird IPv6 von PRADS unterstützt, so dass diese Anforderung als erfüllt gilt.

NFA 7 Anonymisierung
Da PRADS keine Funktion zur Anonymisierung der Ergebnisse vorsieht, ist diese Anforderung nicht erfüllt.

NFA 8 Datenschutz
Hinsichtlich des Datenschutzes ist für PRADS festzuhalten, dass nicht alle Anforderungen erfüllt werden. Wird PRADS in Netzen, in denen sich Dritte befinden, eingesetzt, so besitzt PRADS keine Funktionalität zur Löschung schützenswerter Daten nach der gesetzlichen Höchstspeicherdauer von sieben Tagen. Da jedoch zusätzlich zu der Information der IP-Asset-Beziehung keine weiteren datenschutzrechtlich relevanten Informationen gespeichert werden, ist diese Anforderung als teilweise erfüllt anzusehen.

NFA 9 Gesetzeskonformität
PRADS selbst ist in seinem Funktionsumfang nicht hinsichtlich der Konformität mit deutschen Gesetzen implementiert worden, erfüllt jedoch nach einer ersten Prüfung die gesetzlichen Anforderungen. Kurzgefasst ist dies darin begründet, dass weder geschützte Kommunikation entpackt oder manipuliert noch in Netze Dritter eingedrungen wird.

NFA 10 Transparenz
Da PRADS ein passives Verfahren ist und den Netzverkehr nicht verändert, ist die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist in PRADS-Servern nicht vorgesehen, so dass die Anforderung nicht erfüllt ist.

Tabelle 4.4.: Anforderungscheck PRADS

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	teilweise erfüllt
FA 3	Grenzerfassung	erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	teilweise erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	teilweise erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	teilweise erfüllt
NFA 6	IPv6	erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	teilweise erfüllt
NFA 9	Gesetzeskonformität	erfüllt
NFA 10	Transparenz	erfüllt
NFA 11	Selektion	nicht erfüllt

Zusammengefasst ist PRADS ein nützliches Werkzeug zur Erstellung einer Asset-Datenbank der in einem Netz aktiven Systeme, jedoch können nicht alle Anforderungen dieser Arbeit durch PRADS erfüllt werden. Im Gegensatz zur Analyse von Netflows handelt es sich bei PRADS um einen Netzwerksniffer, der den vollständigen Datenverkehr aufzeichnet und analysiert. Abhängig von der Struktur des betriebenen Netzes ist es notwendig, PRADS entweder auf den genutzten Gateways oder mit Hilfe von Mirror Ports in Switches an speziellen Analyserechnern beziehungsweise innerhalb der Kollisionsdomäne zu erfassen. Diese Anforderung sowie die Funktionsweise von PRADS verstoßen gegen die Anforderungen, zusätzliche Kosten sowie Belastung des Netzes zu vermeiden beziehungsweise diese gering zu halten.

4.2.2. Deep Packet Inspection (DPI)

Eine weitere Möglichkeit Übertragungsinhalte sowie Assets zu detektieren ist der Einsatz von Deep Packet Inspection (DPI). Bei der DPI werden sowohl Datenteil als auch Header eines IP-Pakets analysiert. Die Daten können hierbei innerhalb der Kollisionsdomäne durch Nutzung spezieller Mirror-Ports an Switches oder auch in Routern gesammelt werden. Die so gesammelten Datenpakete lassen sich im Anschluss sowohl zum Inhalt der Übertragung zusammenführen als auch auf Software- und Betriebssystemspuren hin untersuchen. Für die Speicherung und die Untersuchung der gesammelten Pakete wird in der Regel jedoch dedizierte Hardware benötigt.

Zentraler Nachteil der DPI ist, dass auch in kleinen Netzen sehr schnell große Datenmengen anfallen können, da vollständige Datenpakete analysiert werden. Für die beobachteten Systeme ist eine DPI zwar transparent, jedoch kann die Gesamtqualität des Netzes durch den zusätzlichen Traffic negativ beeinträchtigt werden. Einen weiteren Nachteil der DPI stellt die rechtliche Einordnung dar. Zwar ist es laut [Bed09] möglich, zu Sicherheitszwecken DPI einzusetzen, jedoch ist insbesondere auf den Datenschutz zu achten.

Für die Deep Packet Inspection lässt sich festhalten, dass diese kein Verfahren zur Asseterkennung im eigentlichen Sinn, sondern vielmehr eine Funktionsweise zur Analyse von Netzverkehr darstellt. Ferner sind mit dem Einsatz der DPI nicht unerhebliche Ressourcen seitens Rechenleistung, Speicher und Netz notwendig, wodurch Kosten für neue Hardware anfallen können. Abschließend muss festgehalten werden, dass aus juristischer Sicht die Untersuchung fremden Datenverkehrs als fragwürdig einzustufen ist.

FA 1 Heterogenität
Eine Deep Packet Inspection (DPI) ist mit heterogenen Systemen aller Art möglich.

FA 2 Weitläufigkeit
Der Einsatz von DPI in geographisch weitläufigen Netzen ist möglich und kann an allen durchlaufenen Gateways oder Analysepunkten erfolgen.

FA 3 Grenzerfassung
Die DPI in Grenzpunkten zu autonomen Netzen ist möglich.

FA 4 Teilnehmerzahl und Traffic
Die Einsetzbarkeit in großen Netzen mit einer Vielzahl an Teilnehmern und hohem Trafficaufkommen ist als nur bedingt möglich anzusehen, da eine Spiegelung sowie Untersuchung von hohem Datenvolumen mit hohem Aufwand verbunden sein beziehungsweise die bestehende Infrastruktur überfordern kann. Somit ist diese Anforderung als nicht erfüllt zu werten.

FA 5 Asseterkennung
Die Erkennung von Assets ist mit Deep Packet Inspection (DPI) nicht vorgesehen und nur mit Hilfe weiterer Anwendungen möglich, so dass diese Anforderung nicht erfüllt ist.

FA 6 Netze Dritter
Eine Untersuchung in Netze Dritter hinein ist mit Hilfe von Deep Packet Inspection (DPI) nicht möglich, so dass diese Anforderung nicht erüllt wird.

FA 7 Routing
Für die Kompatibilität mit dynamischem Routing ist eine geeignete Wahl des Untersuchungspunktes notwendig, so dass alle Pakete eines Datenstroms den Untersuchungspunkt durchlaufen. Generell ist dies aber möglich, so dass diese Anforderung erfüllt wird.

FA 8 Störungsfreiheit
Hinsichtlich der Störungsfreiheit von Deep Packet Inspection (DPI) kann keine Aussage getroffen werden, da dies von der Nutzungsart abhängt. Somit wird die Anforderung als nicht erfüllt gewertet.

FA 9 Übertragungsinhalte
Eine vollständige Erkennung von Übertragungsinhalten ist mit Hilfe einer Deep Packet Inspection (DPI) möglich, so dass diese Anforderung erfüllt wird.

NFA 1 Verfügbarkeit
Die Durchführung einer Deep Packet Inspection (DPI) hat keinen Einfluss auf die Verfügbarkeit betriebener Dienste und Anwendungen, so dass diese Anforderung erfüllt ist.

NFA 2 Performance
Ist kein Spiegeln des Datenverkehrs notwendig, so wird die Performance nicht negativ beeinflusst und die Anforderung ist erfüllt.

NFA 3 Auswertbarkeit
Durch eine Deep Packet Inspection (DPI) alleine werden lediglich die Paketinhalte auswertbar beziehungsweise der Inhalt der Übertragung erkennbar, eine direkte Auswertbarkeit besteht jedoch nicht. Somit ist diese Anforderung nicht erfüllt.

NFA 4 Kosten
Für eine DPI selbst entstehen keine Lizenzkosten, abhängig von der Art der Umsetzung gegebenenfalls jedoch Kosten für neue Hardware, so dass diese Anforderung teilweise erfüllt wird.

NFA 5 Vorhandene Infrastruktur
Eine Nutzung der vorhandenen Infrastruktur ist nur teilweise möglich, sofern vorhandene Rechenleistung, Speicherkapazität und Bandbreite ausreichend sind. Daher ist diese Anforderung als teilweise erfüllt zu betrachten.

NFA 6 IPv6
Die Nutzung von Deep Packet Inspection (DPI) mit IPv6-Kommunikation ist möglich. Daher gilt die Anforderung als erfüllt.

NFA 7 Anonymisierung
Eine Anonymisierung erfasster Daten ist bei Nutzung von DPI nicht vorgesehen.

NFA 8 Datenschutz
Da die vollständige Unterhaltung analysiert und ausgewertet wird sowie keine Funktionalität zur Einhaltung des Datenschutzes vorgesehen ist, gilt die Anforderung als nicht erfüllt.

NFA 9 Gesetzeskonformität
Das Mitlesen fremder Kommunikation ist als grenzwertig zu betrachten und somit ist diese Anforderung nicht erfüllt.

NFA 10 Transparenz
Da die Deep Packet Inspection (DPI) ein passives Verfahren ist und den Netzverkehr nicht verändert, ist die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist bei der DPI nicht vorgesehen, so dass diese Anforderung nicht erfüllt ist.

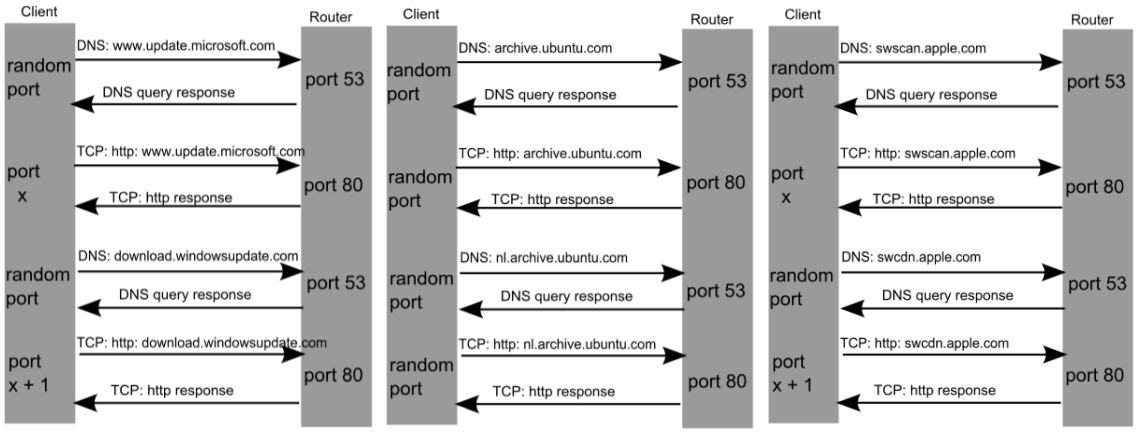
Tabelle 4.5.: Anforderungscheck DPI

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	erfüllt
FA 4	Teilnehmerzahl und Traffic	nicht erfüllt
FA 5	Asseterkennung	nicht erfüllt
FA 6	Netze Dritter	nicht erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	nicht erfüllt
FA 9	Übertragungsinhalte	erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	erfüllt
NFA 3	Auswertbarkeit	nicht erfüllt
NFA 4	Kosten	teilweise erfüllt
NFA 5	Vorhandene Infrastruktur	teilweise erfüllt
NFA 6	IPv6	erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	nicht erfüllt
NFA 10	Transparenz	erfüllt
NFA 11	Selektion	nicht erfüllt

4.2.3. Passive OS detection by monitoring network flows

In seiner Arbeit „Passive OS detection by monitoring network flows“ stellt Siebren Mossel einen Ansatz vor, um mittels Flow-Record-Analyse Rückschlüsse auf genutzte Betriebssysteme zu treffen. Hierfür untersuchte Siebren Mossel, welche Unterschiede bei Systemupdates verschiedener Systeme innerhalb von Flow-Records erkennbar sowie ob unterschiedliche Betriebssystemversionen auf Basis von Flow-Records unterscheidbar sind [Mos10].

In dem für die Untersuchung genutzten Labornetz nutzte Siebren Mossel Windows XP, Mac OS 10.6 sowie Ubuntu 11.04 und erfasste die Verkehrsdaten direkt durch Aufzeichnung der ersten 128 Bytes eines jeden Paketes auf den Quellhosts. Zur Erfassung des entstehenden Hintergrundtraffics und um diesen in zukünftigen Analysen ausblenden zu können, wurden eine Woche lang ohne manuelles Eingreifen die Daten der einzelnen Systeme aufgezeichnet. Im Anschluss wurde der Traffic beim manuellen Abrufen von Updates aufgezeichnet und nachfolgend wurden die gespeicherten Daten mit einem Flow-Printer (YAF Flow) in ein menschenlesbares Format ausgegeben. Bei den genutzten Ports lässt sich feststellen, dass der für die Anfrage der Updates genutzte Port um 1 inkrementiert für das Herunterladen der Updates genutzt wird.



(a) Updating Windows XP (b) Updating Ubuntu 11.04 (c) Updating Mac OS X

Abbildung 4.4.: Verbindungen bei Updates verschiedener Betriebssysteme [Mos10]

Bei der Untersuchung der Flows konnten verschiedene Regelmäßigkeiten festgestellt werden. So wird bei Windows zur Abfrage der verfügbaren Updates eine IP aus dem Bereich von Microsoft (65.52.0.0/14) aufgerufen, während der Download (`download.windowsupdate.com`) zu einer IP des Akamai Technologies CDN aufgelöst wird (vgl. Abbildung 4.4(a)).

Beim Update von Ubuntu lässt sich feststellen, dass zu Beginn immer die URL `archive.ubuntu.com` aufgerufen wird, die auf eine IP aus dem Range (91.189.88.0/21) von Canonical aufgelöst wird. Der Download der Updates erfolgt von einem Update-Server innerhalb des selben Landes wie der Quell-Rechner und nutzt einen zufälligen Port (vgl. Abbildung 4.4(b).

Bei der Installation von Updates unter MacOSX wird zuerst ein zufälliger Port auf dem Host swscan.apple.com aufgerufen, welcher auf eine zu Apple gehörige IP-Adresse auflöst. Von dort wird eine Liste der verfügbaren Updates abgerufen, der Download erfolgt im Anschluss von swcdn.apple.com, was auf eine IP von Akamai Technologies auflöst. Für die Übertragung wird ebenfalls wie bei Windows Updates der Port x+1 genutzt (vgl. Abbildung 4.4(a)).

Zusammengefasst ist es – basierend auf dem Wissen über die aufgerufenen IP-Port-Kombinationen – möglich, Rückschlüsse auf das genutzte Betriebssystem zu ziehen. In einem Test mit 12 Rechnern konnten auf diese Weise Trefferquoten von 83% bis zu 100% erreicht werden. Werden für Updates jedoch eigene Mirrors betrieben, IPv6 genutzt oder kommen Techniken wie Peer-To-Peer zum Einsatz, ist eine Nutzung des so vorgestellten Ansatzes nicht oder nur eingeschränkt möglich. Die für diese Arbeit geforderte Erkennung von eingesetzter Software ist ebenfalls nicht möglich.

FA 1 Heterogenität
Der von Siebren Mossel vorgestellte Ansatz wurde erfolgreich in einer Umgebung mit verschiedenen Betriebssystemen getestet. Somit ist diese Anforderung erfüllt.

FA 2 Weitläufigkeit
Die Erfassung der „angereicherten“ Flow-Records ist in weitläufigen Netzen möglich; entweder geschieht dies in einem zwischengeschalteten Gateway oder in einem zentralen Bordergateway.

FA 3 Grenzerfassung
Das Erfassen der Flow-Records ist in Bordergateways möglich.

FA 4 Teilnehmerzahl und Traffic
Theoretisch ist die vorgestellte Methode auch bei hohem Traffic und vielen Teilnehmern einsetzbar, jedoch wirkt sich hier die Übertragung der ersten 128 Bytes eines jeden Paketes negativ auf die Performance aus. Durch den Autor wurde der Einsatz „im großen Umfang“ ferner nicht getestet, so dass die Eignung vermutlich nur eingeschränkt gegeben ist. Aus diesem Grund ist die Anforderung nur teilweise erfüllt.

FA 5 Asseterkennung
Eine Erkennung verschiedener Betriebssysteme ohne Patchstand / installierte Software wurde durch den Autor erfolgreich getestet. Die in Anforderung FA 5 verlangte Erkennung von installierter Software, Versionsstand und Betriebssystemen ist nicht vollständig gegeben, so dass die Anforderung teilweise erfüllt ist.

FA 6 Netze Dritter
Da eine passive Analyse des durchgeleiteten Traffics erfolgt, ist die Kompatibilität mit Netzen Dritter möglich.

FA 7 Routing
So lange der zu analysierende Traffic den Analyseserver (oder ein Gateway, das die Flow-Records erfasst) durchläuft, ist die Analyse möglich und die Anforderung somit erfüllt.

FA 8 Störungsfreiheit
Da lediglich eine passive Analyse des anfallenden Datenverkehrs erfolgt, ist die Anforderung der Störungsfreiheit erfüllt.

FA 9 Übertragungsinhalte
Systemseitig ist eine Detektion der Übertragungsinhalte nicht vorgesehen, so dass diese Anforderung nicht erfüllt ist.

NFA 1 Verfügbarkeit
Durch die Analyse erfolgt keine negative Beeinflussung der Verfügbarkeit betriebener Dienste und Anwendungen, so dass diese Anforderung erfüllt ist.

NFA 2 Performance
Abhängig von der Anzahl des Gesamttraffics im Netz kann bei Übertragung der ersten 128 Bytes eines jeden Paketes eine Störung des Netzes erfolgen. Werden die Daten lediglich im Gateway erfasst und dort ausgewertet, erfolgt keine negative Beeinflussung. Die Anforderung ist daher teilweise erfüllt.

NFA 3 Auswertbarkeit
Die Resultate des in der Arbeit von Siebren Mossel vorgestellten Verfahrens ermöglichen eine weitere Auswertung. Somit ist diese Anforderung erfüllt.

NFA 4 Kosten
Lizenzen oder ähnliche Kosten fallen für die vorgestellte Lösung nicht an, so dass diese Anforderung erfüllt ist.

NFA 5 Vorhandene Infrastruktur
Für die Erfassung der Daten lässt sich die vorhandene Infrastruktur, genauer gesagt die bestehenden Gateways nutzen, so dass diese Anforderung erfüllt ist.

NFA 6 IPv6
Eine Nutzung des vorgestellten Ansatzes mit IPv6 wurde bisher nicht getestet, wobei auf Grund der Detektion mittels der IPv4 Adressen verschiedener Hersteller eine Kompatibilität als nicht gegeben anzusehen ist. Somit ist diese Anforderung nicht erfüllt.

NFA 7 Anonymisierung
Da keine Funktion zur Anonymisierung der Ergebnisse vorgesehen ist, wird diese Anforderung nicht erfüllt.

NFA 8 Datenschutz
Für die Anforderung des Datenschutzes ist festzuhalten, dass nicht alle Anforderungen erfüllt werden. Erfolgt eine Analyse in Netzen Dritter, so fehlt die Funktionalität zur Löschung schützenswerter Daten nach der gesetzlichen Hochstspeicherdauer von sieben Tagen. Da jedoch zusätzlich zur Information der IP-Asset-Beziehung keine weiteren datenschutzrechtlich relevanten Informationen gespeichert werden, kann diese Anforderung als teilweise erfüllt angesehen werden.

NFA 9 Gesetzeskonformität
Da keine Manipulation, jedoch aber eine partielle Analyse der Übertragungsinhalte erfolgt, ist die vorgestellte Lösung als grenzwertig einzustufen. Auf Grund der lediglich partiellen Inhaltsanalyse, welche in der Regel keine Rückschlüsse auf Übertragungsinhalte ermöglicht, lässt sich die Anforderung als teilweise erfüllt ansehen.

NFA 10 Transparenz
Da der vorgestellte Ansatz ein passives Verfahren ist und somit den Netzverkehr nicht verändert, wird die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist in der vorgestellten Lösung nicht vorgesehen, so dass die Anforderung nicht erfüllt ist.

Tabelle 4.6.: Anforderungscheck Passive OS detection by monitoring network flows

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	teilweise erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	teilweise erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	erfüllt
NFA 6	IPv6	nicht erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	teilweise erfüllt
NFA 9	Gesetzeskonformität	teilweise erfüllt
NFA 10	Transparenz	erfüllt
NFA 11	Selektion	nicht erfüllt

4.2.4. Passive Detektion von Betriebssystem und installierter Software mittels Flow-Records

In ihrer Publikation „Passive Detektion von Betriebssystem und installierter Software mittels Flow-Records“ stellen Michael Grabatin und Felix von Eye einen Ansatz vor, mit Hilfe von Flow-Records Schlüsse auf eingesetztes Betriebssystem wie auch genutzte Software zu ziehen [GvE16, GvE05]. Hierbei beschreiben die Autoren ausführlich die Möglichkeiten wo und wie sich Flow-Records sammeln lassen, in wie weit Flow-Records genutzt werden können und begründen, warum das Wissen über die im Netz befindlichen Assets notwendig ist.

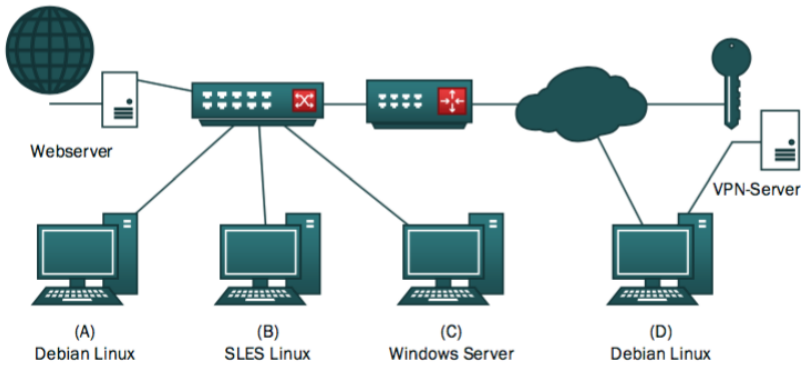


Abbildung 4.5.: Aufbau der Testumgebung [GvE16]

Für Ihre Arbeit nutzten Michael Grabatin und Felix von Eye die in Abbildung 4.5 beschriebene Testumgebung, in der auf einem Webserver Dateien im Größenbereich von 1 MB bis 50 MB zum Download bereitgestellt wurden. Des Weiteren wurden auf dem Webserver die zu untersuchenden Flow-Records gesammelt und die Hop-Zahl des Traffics testweise unter Nutzung eines VPNs erhöht. Durch die verschiedenen Testsysteme wurden die Daten 25 mal heruntergeladen und die gesammelten Flow-Records ausgewertet. Hierbei wurde festgestellt, dass mit Erhöhung der Distanz zwischen Quellsystem und Zielsystem die Genauigkeit abnimmt, jedoch auch unter Nutzung des VPN noch 61% richtig klassifiziert werden konnten, wobei eine Analyse der durch den Internet Explorer heruntergeladenen Daten durch Nutzung des identischen Ports für mehrere Downloads erschwert beziehungsweise verhindert wurde. Durch weitere Optimierung der Analyse wie beispielsweise der Korrektur des durch Hops entstehenden Overheads der Paketheader konnte das Ergebnis der Klassifizierung im Verlauf weiter optimiert werden.

Als Abschluss ihrer Arbeit diskutieren die Autoren die notwendigen Grundlagen, wie verlässliche Fingerprints von Updates sowie das Wissen über aufgebaute Verbindungen gesammelt werden können, um eine Klassifikation von Betriebssystemen und eingesetzter Software zu ermöglichen. Jedoch liefern sie noch kein vollständiges Konzept zur Erkennung genutzter Software sowie Betriebssysteme.

FA 1 Heterogenität
Der Einsatz in heterogenen Netzen wurde durch die Autoren erfolgreich getestet.

FA 2 Weitläufigkeit
Basierend auf dem bei der Datenerfassung genutzten Verfahren ist der Einsatz in weitläufigen Netzen möglich.

FA 3 Grenzerfassung
Durch das für die Erzeugung von Flow-Records genutzte Verfahren ist ebenfalls eine Datenerfassung in Grenzgateways möglich. Somit ist diese Anforderung erfüllt.

FA 4 Teilnehmerzahl und Traffic
Das vorgestellte Verfahren ist auf Grund seiner Leichtigkeit in Netzen mit vielen Teilnehmern und hohem Trafficaufkommen einsetzbar.

FA 5 Asseterkennung
Eine Erkennung von Assets basierend auf regulärer Kommunikation ist mit diesem Ansatz nicht möglich. Bewiesen wurde jedoch, dass sich Flow-Records ausreichend unterscheiden, um eine Auswertung vornehmen zu können. Dennoch ist diese Anforderung nicht erfüllt.

FA 6 Netze Dritter
Die Erfassung an Grenzpunkten zu Netzen Dritter ist möglich und somit die Erfassung aller nach außen hin kommunizierenden Systeme.

FA 7 Routing
Solange den Flow-Recorder die zu analysierenden Daten durchlaufen, ist die Kompatibilität mit komplexem Routing gegeben und diese Anforderung erfüllt.

FA 8 Störungsfreiheit
Durch die passive Analyse und Erfassung der Flow-Records erfolgt keine Störung bestehender Systeme.

FA 9 Übertragungsinhalte
Eine Erkennung von unbekannten Übertragungsinhalten ist durch das vorgestellte Verfahren nicht möglich.

NFA 1 Verfügbarkeit
Das vorgestellte Verfahren schränkt die Verfügbarkeit betriebener Dienste und Anwendungen durch die Analyse nicht ein, so dass diese Anforderung als erfüllt gilt.

NFA 2 Performance
Da das durch dieses Verfahren übertragene Datenvolumen derart gering ist, erfolgt keine negative Beeinflussung der Netzperformance.

NFA 3 Auswertbarkeit
Eine Auswertung und Weiterverarbeitung der durch das Verfahren gewonnenen Daten ist möglich, jedoch existiert hierfür noch kein geeignetes Format, so dass diese Anforderung nur teilweise erfüllt ist.

NFA 4 Kosten
Der Einsatz der vorgestellten Lösung erfordert weder Lizenzkosten noch Kosten für neue Hardware, so dass diese Anforderung als erfüllt gilt.

NFA 5 Vorhandene Infrastruktur
Die Erfassung von Flow-Records ist sowohl durch moderne Netzhardware als auch durch günstig aufsetzbare Flow-Recorder möglich. Daher gilt diese Anforderung als erfüllt.

NFA 6 IPv6
Der Einsatz mit IPv6 wurde bisher nicht getestet, ist aber auf Grund des Verfahrens als möglich einzustufen, so dass diese Anforderung als erfüllt zu betrachten ist.

NFA 7 Anonymisierung
Da keine Funktion zur Anonymisierung der Ergebnisse vorgesehen ist, wird diese Anforderung nicht erfüllt.

NFA 8 Datenschutz
Da keine Funktionalität zur Löschung schützenswerter Daten nach Ablauf der Höchstspeicherdauer existiert, somit die Daten weiterhin in den aufgezeichneten Flow-Records vorhanden sind, gilt diese Anforderung analog zu den vorhergehend genannten Verfahren als teilweise erfüllt.

NFA 9 Gesetzeskonformität
Da kein Entpacken oder Verändern der Netzpakete erfolgt, ist die Anforderung als erfüllt anzusehen.

NFA 10 Transparenz
Da der vorgestellte Ansatz ein passives Verfahren ist, welches den Netzverkehr nicht verändert, wird die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist in der vorgestellten Lösung nicht vorgesehen, so dass die Anforderung nicht erfüllt ist.

Tabelle 4.7.: Anforderungscheck Passive Detektion von Betriebssystem und installierter Software mittels Flow-Records

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	erfüllt
FA 4	Teilnehmerzahl und Traffic	erfüllt
FA 5	Asseterkennung	nicht erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	erfüllt
NFA 3	Auswertbarkeit	teilweise erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	erfüllt
NFA 6	IPv6	erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	teilweise erfüllt
NFA 9	Gesetzeskonformität	erfüllt
NFA 10	Transparenz	erfüllt
NFA 11	Selektion	nicht erfüllt

4.2.5. Identifying Operating System Using Flow-based Traffic Fingerprinting

Tomáš Jirsík und Pavel Čeleda untersuchen in ihrer Arbeit „Identifying Operating System Using Flow-based Traffic Fingerprinting“ die Detektierbarkeit von Betriebssystemen durch Flow-basierte Analysen. Hierfür definieren sie zu Beginn die Anforderungen, dass ein entsprechenden Analysetool einfach ausrollbar, auf passiver Detektion basierend und sehr performant sein muss, um auch größere Netze untersuchen zu können [Jv14] .

Für die Betriebssystemanalyse werden mit den Flows auch TTL, SYN-Paketgröße, initiale TCP-Fenstergröße sowie das User-Agent-Feld des Headers des HTTP-Protokolls genutzt. Für die anschließende Betriebssystemerkennung erfolgt ein Vergleich mit einer Finger-Print-Datenbank, wobei die Optimierung des Verfahrens ausdrücklich in den Ausblick für weitere Arbeiten verschoben wird. Wurde ein Betriebssystem erkannt, so wird die gewonnene Information den Flows hinzugefügt.

Die Autoren betonen ausdrücklich, dass ihre Arbeit der Ausarbeitung eines Konzepts dient, welches einen deutlichen Performancegewinn im Vergleich zu bestehenden Softwarelösungen bietet. Das so vorgestellte Framework ist im Rahmen dieser Arbeit jedoch nicht nutzbar, da hierfür Paketinhalte entpackt und analysiert werden müssten, welche in den zu analysierenden Flow-Records nicht vorhanden sind. Ebenfalls spricht gegen die Nutzung des Frameworks, dass lediglich eine Analyse von HTTP-Traffic möglich ist sowie die genutzte Software nicht detektiert werden kann.

FA 1 Heterogenität
Das durch die Autoren vorgestellte Konzept ist auf den Einsatz in heterogenen Systemen ausgelegt.

FA 2 Weitläufigkeit
Eine Einschätzung der Anwendbarkeit ist schwierig, da es sich um ein reines Konzeptpapier handelt. Basierend auf der Beschreibung der Autoren sollte der Einsatz in weitläufigen Netzen möglich sein, so dass diese Anforderung als erfüllt betrachtet werden kann.

FA 3 Grenzerfassung
Die Erfassung der beschriebenen Daten ist generell in allen durchlaufenen Gateways und daher auch in Grenzpunkten möglich.

FA 4 Teilnehmerzahl und Traffic
Da das im Konzept beschriebene Verfahren nicht nur die Flow-Records sondern auch eine Vielzahl weiterer Informationen auswertet, ist es als zweifelhaft anzusehen, ob dieses Konzept in Netzen mit hohem Trafficaufkommen und vielen Teilnehmern umsetzbar ist. Da dies jedoch auch nicht ausgeschlossen werden kann, wird diese Anforderung als teilweise erfüllt gewertet.

FA 5 Asseterkennung
Zwar bietet das vorgestellte Konzept die Möglichkeit, Betriebssysteme zu erkennen, jedoch lassen sich nicht alle durch die Anforderung verlangten Daten erfassen, so dass diese Anforderung teilweise erfüllt ist.

FA 6 Netze Dritter
Die Analyse von vorhandenen Hosts in Netzen Dritter ist generell möglich, so lange der entstehende Traffic einen entsprechenden Analysesserver durchläuft. Somit gilt diese Anforderung als erfüllt.

FA 7 Routing
So lange der Analyeserver durchlaufen wird, ist die Kompatibilität mit komplexem Routing gegeben.

FA 8 Störungsfreiheit
Da das Verfahren rein passiv arbeitet, ist die Störungsfreiheit gegeben und die Anforderung erfüllt.

FA 9 Übertragungsinhalte
Eine Erkennung von Übertragungsinhalten ist durch das Tool nicht vorgesehen, so dass diese Anforderung als nicht erfüllt gillt.

NFA 1 Verfügbarkeit
Die Verfügbarkeit betriebener Dienste und Anwendungen wird durch die Analyse nicht eingeschränkt. Somit ist die Anforderung erfüllt.

NFA 2 Performance
Diese Anforderung gilt als erfüllt, da die Performance des Netzes durch die im Konzept beschriebene Analyse nicht beeinträchtigt wird.

NFA 3 Auswertbarkeit
Bezüglich der Auswertbarkeit wurde durch die Autoren keine konkrete Aussage getroffen, so dass zu dieser Anforderung keine Bewertung erfolgen kann. Daher gilt diese Anforderung als nicht erfüllt.

NFA 4 Kosten
Das beschriebene Verfahren kommt ohne Lizenzkosten oder Kosten für weitere Hardware aus, so dass diese Anforderung als erfüllt gilt.

NFA 5 Vorhandene Infrastruktur
Da die Analyse lediglich in vollwertigen Rechnern wie beispielsweise Gateways möglich ist, die Datenerfassung jedoch nicht über normale Netzhardware erfolgen kann, gilt diese Anforderung als teilweise erfüllt.

NFA 6 IPv6
Zur Nutzung mit IPv6 erfolgte keine Bewertung durch die Autoren, so dass die Anforderung als nicht erfüllt gewertet wird.

NFA 7 Anonymisierung
Da keine Funktion zur Anonymisierung der Ergebnisse vorgesehen ist, wird diese Anforderung nicht erfüllt.

NFA 8 Datenschutz
Da zur Speicherung und Speicherdauer keine Aussage durch die Autoren erfolgt, gilt diese Anforderung als nicht erfüllt.

NFA 9 Gesetzeskonformität
Da kein Entpacken oder Verändern der Netzpakete erfolgt, ist die Anforderung als erfüllt anzusehen.

NFA 10 Transparenz
Da der vorgestellte Ansatz ein passives Verfahren ist, welches den Netzverkehr nicht verändert, wird die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist in der vorgestellten Lösung nicht vorgesehen, so dass die Anforderung nicht erfüllt ist.

Tabelle 4.8.: Anforderungscheck Identifying Operating System Using Flow-based Traffic Fingerprinting

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	erfüllt
FA 4	Teilnehmerzahl und Traffic	teilweise erfüllt
FA 5	Asseterkennung	teilweise erfüllt
FA 6	Netze Dritter	erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	erfüllt
NFA 2	Performance	erfüllt
NFA 3	Auswertbarkeit	nicht erfüllt
NFA 4	Kosten	erfüllt
NFA 5	Vorhandene Infrastruktur	teilweise erfüllt
NFA 6	IPv6	nicht erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	erfüllt
NFA 10	Transparenz	erfüllt
NFA 11	Selektion	nicht erfüllt

4.2.6. Bewertung passiver Verfahren

Als Beispiele für passive Analyseverfahren und Asseterkennungssysteme wurden in diesem Abschnitt verschiedene Konzepte und Methoden vorgestellt und hinsichtlich der Erfüllung der Anforderungen dieser Arbeit untersucht. Allen Verfahren ist gemeinsam, dass diese weder aktiv kommunizieren noch den anfallenden Datenverkehr verändern. Abhängig von Art und Umsetzung der vorgestellten Konzepte reichten diese für die Netzteilnehmer von weder spürbar (durch Störungen) noch an den Hosts detektierbar bis hin zu starken negativen Beeinflussungen der Netzteilnehmer. Funktionell war es den meisten vorgestellten Verfahren möglich, das eingesetzte Betriebssystem zu detektieren und dabei die vorhandene Infrastruktur zu nutzen. Zentrale Schwachstellen waren jedoch zum Einen, dass meist weder eingesetzte Software noch Versionsstände detektiert werden konnten, zum Anderen, dass nur aktiv im Netz kommunizierende Systeme erkannt wurden.

Durch Abbildung 4.6 wird verdeutlicht, dass die in Kapitel 4.2 vorgestellten Verfahren und Tools die im Rahmen dieser Thesis erarbeiteten Anforderungen auch kombiniert nicht oder nur teilweise erfüllen.

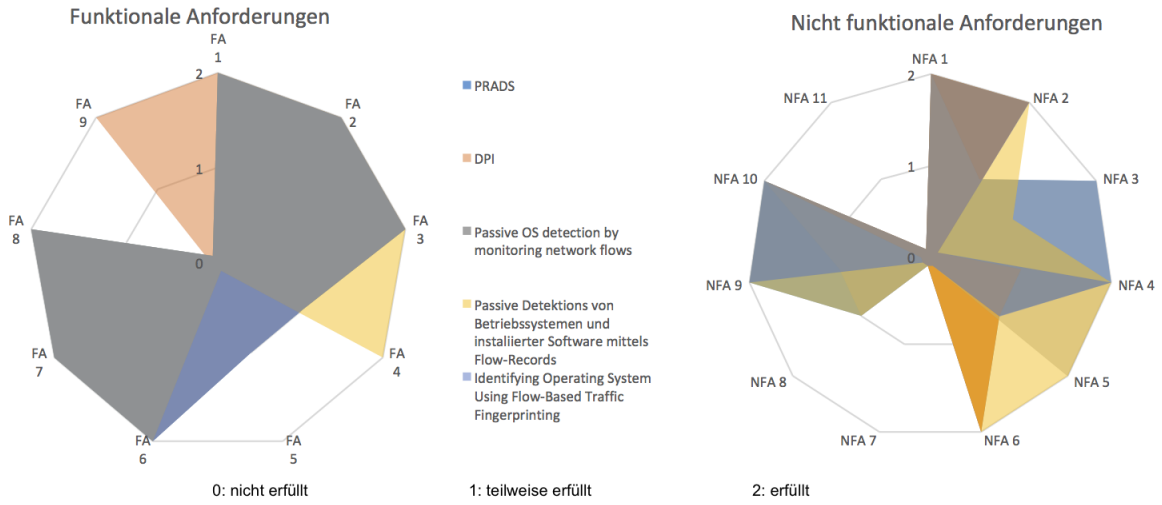


Abbildung 4.6.: Visualisierung der Anforderungserfüllung durch passive Verfahren

4.3. Hybride Verfahren

In ihrer Arbeit „Automated Service Discovery for Enterprise Network Management“ untersuchen William Tu, Priya Thangaraj, Jui-hao Chiang und Tzicker Chiueh die Möglichkeiten, die sich in einem Netz befindlichen Assets mittels Kombination aktiver und passiver Verfahren zu detektieren und die generierte Liste aktuell zu halten. Zentrales Ziel des entwickelten hybriden Verfahrens ist es, möglichst kostensparend eine möglichst aktuelle, vollständige und sich selbst aktualisierende Liste der sich im Netz befindlichen Assets zu generieren [TThCcC09].

Zur Erstellung dieser Liste wird initial mittels dem aktiven Scanverfahren `Nmap` ein vollständiger Scan des Netzes durchgeführt und basierend auf diesem Scan ermittelt, welches System sich topologisch an welcher Stelle des Netzes (hinter welchem Switch) befindet. Im weiteren Verlauf werden Veränderungen mittels Analyse der an den Switches anfallenden Flow-Records (Source IP, Source Port, Ziel IP, Ziel Port, Protokoll) überwacht. Flows bekannter (bereits gescannter) Dienste werden anschließend identifiziert, andernfalls als unbekannt getaggt und für die weitere Untersuchung in einer Zu-Scannen-Datenbank gespeichert. Die so entstandene Datenbank wird periodisch abgearbeitet und die erfassten Dienste analysiert, um einen vollständigen Scan des Netzes zu vermeiden. Um verwaiste Datenbankeinträge in der Liste gefundener Dienste ausschließen zu können, wird diese Liste regelmäßig durch `Nmap` auf nicht mehr existierende Dienste hin untersucht.

In der Evaluation ihrer Lösung simulieren die Autoren ein Netz aus fünf Switches, einem Gateway und ca. 20 Hosts, die verschiedene Anwendungen wie http, https, ntp, nfs, ssh, cvsperver, Microsoft-ds, netbios-ssn, ipp, dns, ldap, telnet, rpcbind, und vpn betreiben. Der initiale vollständige Scan dieses Netzes dauert ca. 15 bis 20 Minuten. Diese Dauer zeigt die Einschränkung des Verfahrens auf, da es zu Beginn einen vollständigen aktiven Scan des Netzes erfordert und so für dynamische und große Netze, wie es für diese Arbeit gefordert wird, nicht anwendbar ist. Des Weiteren wird die Anforderung der Transparenz für User hier ebenfalls nicht erfüllt.

FA 1 Heterogenität
Das Tool ist für den Einsatz in heterogenen Netzen vorgesehen und durch die Autoren entsprechend getestet, so dass die Anforderung als erfüllt gilt.

FA 2 Weitläufigkeit
Sofern keine zu starke Verzögerung (Ping) und somit lange Laufzeiten durch die Weitläufigkeit entstehen, gilt die Anforderung als erfüllt.

FA 3 Grenzerfassung
Die Anforderung an Grenzpunkten zu autonomen Netzen einsetzbar zu sein, wird nur teilweise erfüllt, da hierfür immer ein aktiver Scan in das Netz hinein notwendig ist. Sofern dies nicht möglich ist, besteht keine Möglichkeit die Anforderung zu erfüllen.

FA 4 Teilnehmerzahl und Traffic
Ein Einsatz in Netzen mit hoher Teilnehmerzahl ist als nicht möglich anzusehen, da die Scandauer mit Zunahme der Hosts erheblich wächst und das Tool so unbrauchbar macht.

FA 5 Asseterkennung
Die Erkennung von Assets – inklusive betriebener Dienste – ist durch das Tool vorgesehen und getestet, so dass die Anforderung als erfüllt betrachtet werden kann.

FA 6 Netze Dritter
Falls ein Scan in Netze Dritter hinein nicht möglich ist, kann diese Anforderung nicht erfüllt werden. Da das beschriebene Verfahren ferner Zugriff auf die Hardware des Netzes benötigt, ist die Anforderung als nicht erfüllt anzusehen.

FA 7 Routing
Solange der Analyseserver alle Hosts erreichen kann, ist die Kompatibilität mit komplexem Routing gegeben, und damit diese Anforderung als erfüllt anzusehen.

FA 8 Störungsfreiheit
Der notwendige aktive Initialscan kann Störungen verursachen, so dass diese Anforderung als teilweise erfüllt angesehen werden kann.

FA 9 Übertragungsinhalte
Eine Erfassung von Übertragungsinhalten ist nicht vorgesehen, so dass diese Anforderung nicht erfüllt wird.

NFA 1 Verfügbarkeit
Der notwendige aktive Scan kann sich negativ auf die Verfügbarkeit betriebener Dienste und Anwendungen auswirken, so dass diese Anforderung als teilweise erfüllt anzusehen ist.

NFA 2 Performance
Der notwendige vollständige aktive Netzscan kann sich stark negativ auf die Performance des Netzes auswirken. Daher gilt diese Anforderung als nicht erfüllt.

NFA 3 Auswertbarkeit
Da alle Ergebnisse in einer zentralen Datenbank erfasst werden, ist eine Auswertung und Weiterverarbeitung durch Nutzung dieser Datenbank einfach möglich. Somit ist diese Anforderung erfüllt.

NFA 4 Kosten
Zwar fallen keine Lizenzkosten für die vorgestellte Lösung an, jedoch kann es notwendig sein, zur Verbesserung der Performance weitere Hardware zu beschaffen. Somit ist diese Anforderung nur teilweise erfüllt.

NFA 5 Vorhandene Infrastruktur
Das vorgestellte Konzept baut auf der Nutzung der vorhandenen Infrastruktur, genauer gesagt der Netzinfrastruktur, auf. Nach der initialen Netzanalyse über einen Netzscanner wie `Nmap` wird nah an den zu überwachenden Systemen mittels der Netzhardware weiter analysiert. Da für den initialen Scan jedoch dedizierte Rechner benötigt werden, ist die Anforderung teilweise erfüllt.

NFA 6 IPv6
Das vorgestellte Konzept ist generell mit IPv6 kompatibel, so dass die Anforderung als erfüllt zu betrachten ist.

NFA 7 Anonymisierung
Da keine Funktion zur Anonymisierung der Ergebnisse vorgesehen oder implementiert ist, wird diese Anforderung nicht erfüllt.

NFA 8 Datenschutz
Da die Datenbank auf längerfristige Speicherung der Daten ausgelegt ist und somit die Höchstspeicherdauer überschritten wird sowie keine Funktion zur Löschung oder Anonymisierung existiert, wird diese Anforderung nicht erfüllt.

NFA 9 Gesetzeskonformität
Ein Entpacken oder Verändern der Netzpakete erfolgt nicht, jedoch wird ein Scan in Netze beziehungsweise auf Systemen Dritter durchgeführt. Daher ist die Anforderung als teilweise erfüllt anzusehen.

NFA 10 Transparenz
Da der vorgestellte Ansatz ein aktives Verfahren für die initiale Erstellung des Netzplans sowie für die Erfassung von Änderungen nutzt, gilt diese Anforderung als nicht erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist in der vorgestellten Lösung nicht vorgesehen, so dass die Anforderung nicht erfüllt ist.

Tabelle 4.9.: Anforderungscheck Automated Service Discovery for Enterprise Network Management

ID	Schlagwort	Bewertung
FA 1	Heterogenität	erfüllt
FA 2	Weitläufigkeit	erfüllt
FA 3	Grenzerfassung	nicht erfüllt
FA 4	Teilnehmerzahl und Traffic	nicht erfüllt
FA 5	Asseterkennung	erfüllt
FA 6	Netze Dritter	nicht erfüllt
FA 7	Routing	erfüllt
FA 8	Störungsfreiheit	teilweise erfüllt
FA 9	Übertragungsinhalte	nicht erfüllt
NFA 1	Verfügbarkeit	teilweise erfüllt
NFA 2	Performance	nicht erfüllt
NFA 3	Auswertbarkeit	erfüllt
NFA 4	Kosten	teilweise erfüllt
NFA 5	Vorhandene Infrastruktur	teilweise erfüllt
NFA 6	IPv6	erfüllt
NFA 7	Anonymisierung	nicht erfüllt
NFA 8	Datenschutz	nicht erfüllt
NFA 9	Gesetzeskonformität	teilweise erfüllt
NFA 10	Transparenz	nicht erfüllt
NFA 11	Selektion	nicht erfüllt

4.3.1. Bewertung hybrider Verfahren

Mangels weiterer hybrider Verfahren wurde in Kapitel 4.3 lediglich ein Verfahren vorgestellt. Generell ist festzuhalten, dass der Ansatz einer hybriden Umsetzung unter Nutzung der bestehenden Netzinfrastruktur ein sehr gutes Ergebnis ermöglichen kann. Wesentliche Schwachstelle des vorgestellten Ansatzes ist jedoch die hohe Dauer des initialen Scans, durch die es im Einsatz in größeren Netzen dazu kommen kann, dass bis zum Ende des Scans Hosts nicht mehr aktiv sind. Eine Kombination mit anderen Verfahren oder die Nutzung einer verteilten Analyse könnte hier zu besseren Ergebnissen führen.

Durch Abbildung 4.7 wird verdeutlicht, dass das in Kapitel 4.3 vorgestellte Verfahren die im Rahmen dieser Thesis erarbeiteten Anforderungen nicht oder nur teilweise erfüllen.

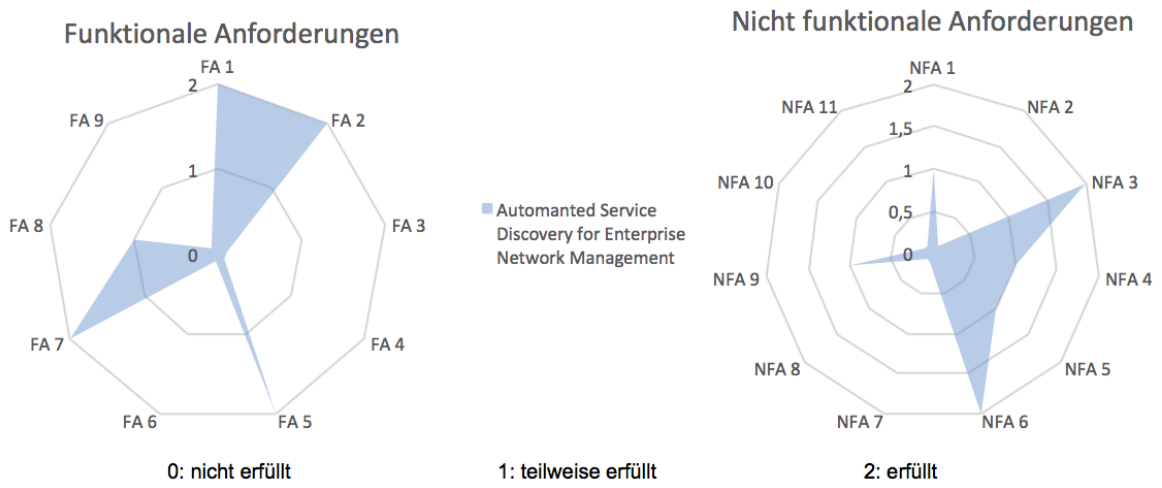


Abbildung 4.7.: Visualisierung der Anforderungserfüllung durch hybride Verfahren

4.4. Bewertung themenverwandter Arbeiten

In Kapitel 4 wurden verschiedene thematisch verwandte Arbeiten, die aktive, passive oder hybride Detektionsverfahren nutzen, hinsichtlich der Gesamtanforderungen dieser Arbeit validiert. Dabei wurde festgestellt, dass keines der untersuchten Verfahren vollständig geeignet ist, alle Anforderungen zu erfüllen.

Tendenziell bietet es sich an sowohl aktive als auch passive Verfahren zu kombinieren. Aktive Verfahren ermöglichen auch aktuell nicht kommunizierende Dienste und Systeme zu erkennen, passive Verfahren bieten hingegen den Vorteil, dass durch diese, insbesondere bei Nutzung kommunikationsarmer Techniken, keine Belastung der betriebenen Dienste erfolgt.

Des Weiteren ist es sinnvoll, eine Differenzierung nach Einsatzgebiet vorzunehmen, da Netze wie beispielsweise Rechenzentren, in denen sich Konfigurationen nur selten ändern, einen geringeren Bedarf an Vollscans haben und entsprechend nur Änderungen überwacht werden müssen. Im Gegensatz hierzu erfordert ein Netz wie beispielsweise das Münchner Wissenschaftsnetz (MWN) mit seinen dynamisch vergebenen IP-Adressen und den oft nicht lange verbundenen Clients ein stetiges Monitoring.

Für den Einsatz in Rechenzentren, in denen potentiell eine vollständige Asset-Datenbank fehlt, aber Assets eine längere Lebensdauer besitzen, bietet sich eine Kombination analog zum vorgestellten hybriden Ansatz an. Hierfür lässt sich das Tool `Nmap` nutzen, um eine Datenbank der bestehenden Assets zu erzeugen. Zur Reduzierung der Laufzeit lässt sich der Scan des Netzes auf verschiedene Scanserver (virtuelle kostengünstige Maschinen in den jeweiligen Subnetzen) verteilen. Anschließend kann mit Hilfe des von Felix von Eye et al. vorgestellten Ansatzes der Flow-basierten Betriebssystemerkennung beobachtet werden, ob sich Assets atypisch verhalten und erneut aktiv untersucht werden müssen. Da bei einem derartigen Ansatz die Anzahl der aktiven Scans auf ein Minimum reduziert wird und bestehende Netz-Infrastruktur zur Erfassung der Flow-Records genutzt werden kann, werden mit großer Wahrscheinlichkeit gute Resultate hinsichtlich der Anforderungserfüllung erzielt.

Im Gegensatz zu der relativ statischen Umgebung in Rechenzentren ändern sich beispielsweise in einem Hochschulnetz oder Unternehmensnetz die angeschlossenen Systeme häufiger, werden heruntergefahren oder an anderen Orten wieder mit dem Netz verbunden. Somit ist eine Lösung, welche auf einer statischen Datenbank aufbaut, in diesem Szenario nicht einsetzbar. Hier bietet es sich an, eine Kombination von `PRADS` in den genutzten Vermittlungsgateways nahe am Zielhost, einer Datenbank, einem eigenen Auswertungs- und Steuerungstool und einem aktivem Untersuchungstool wie `Nmap` einzusetzen. So wäre es möglich mittels `PRADS` in Echtzeit Daten der Subnetze zu erfassen und durch das Auswertungstool einsammeln zu lassen. Anschließend lassen sich diese Daten mit einem Timestamp in einer Datenbank sammeln, wobei das Auswertungstool für die Einhaltung des Datenschutzes sorgt und entsprechend die Daten nach Ablauf löscht. Hosts, welche auffällig lange verfügbar sind, können als Server angesehen werden und dann durch `Nmap` hinsichtlich der betriebenen Dienste weiter untersucht werden. Mit diesem Konzept ließe sich eine kostengünstige Lösung abbilden, welche die meisten Anforderungen erfüllt, jedoch nicht transparent ist.

5. Konzeptaufbau und -Erläuterung des FRF-Tools

Ziel dieser Arbeit ist mit Hilfe von Flow-Records aussagekräftige Informationen über eingesetzte Betriebssysteme, betriebene Dienste sowie die installierte Software von detektierten Hosts erhalten zu können. Hierbei bieten Flow-Records den elementaren Vorteil der Leichtgewichtigkeit und erfüllen somit etliche der Gesamtanforderungen aus Kapitel 3. Ebenfalls sei angemerkt, dass Flow-Records auch erlauben, vielen der in Kapitel 4 aufgezeigten Problemen der aktiven und passiven Verfahren zu begegnen.

Das in diesem Kapitel beschriebene Konzept des FRF-Tools besteht aus drei Prozessschritten (vgl. Abbildung 5.1).

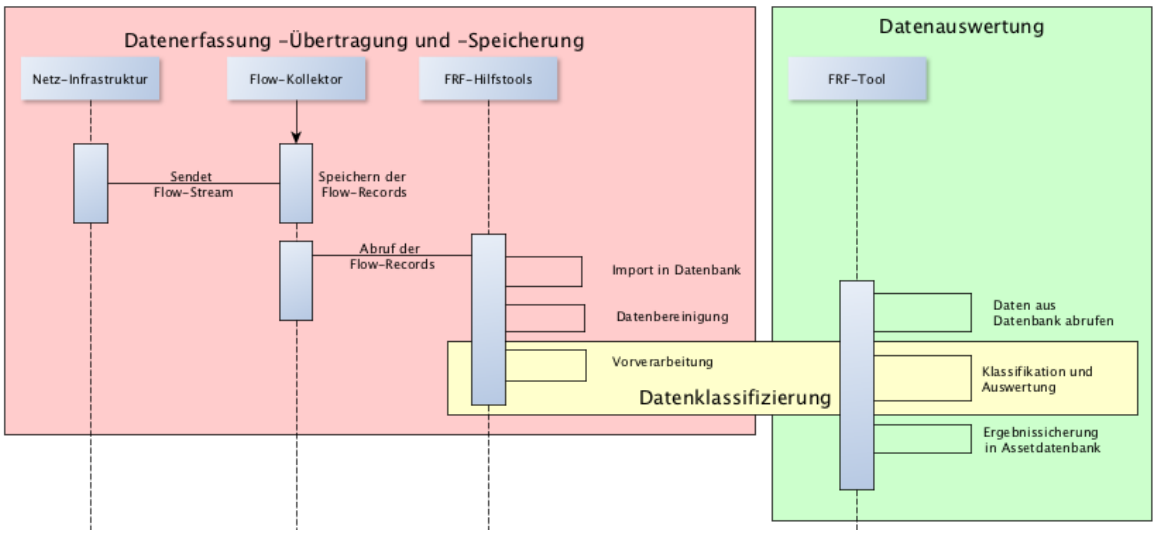


Abbildung 5.1.: Visualisierung der Hauptbereiche des Konzepts

Der erste Prozessschritt besteht aus der Datenerfassung, der Übertragung hin zum FRF-Tool sowie der Speicherung dieser erfassten Daten durch das FRF-Tool. Hierbei wird bei der Datenerfassung insbesondere auf die Fragestellung wo und wie Daten erfasst werden können eingegangen.

Die Datenklassifizierung, welche der Vorverarbeitung dient, stellt den zweiten Prozessschritt dar. An dieser Stelle werden grundlegende Begriffe definiert sowie die Aufbereitung der gesammelten Flow-Records vor der eigentlichen Verarbeitung erläutert.

Der letzte Prozessschritt beschreibt, wie die eingelesenen und durch die Klassifizierung aufbereiteten Daten innerhalb des FRF-Tools durch heuristische Verfahren ausgewertet werden. Abschließend wird hier die Speicherung der gewonnenen Resultate in einer Asset Datenbank vorgenommen.

5.1. Datenerfassung, -Übertragung und -Speicherung

Das Generieren von Flow-Records ist sowohl in verschiedenen Netzkomponenten als auch in verschiedenen Bereichen des Netzes möglich. Zu diesen Netzkomponenten zählen unter anderem Switches, Controller oder auch Router. Die Netzbereiche, die in Abbildung 5.2 visualisiert werden, unterscheiden sich in der Regel durch die Distanz zum Anwender sowie die Anzahl der Netzteilnehmer und -Komponenten. Betrachtet man an dieser Stelle das Beispiel eines Hochschulnetzes, so ist es möglich Flow-Records direkt innerhalb eines Arbeitsgruppennetzes, innerhalb von Institutsnetzen oder auch in autonomen Netzen sowie dem Internet zu erfassen. Festzuhalten ist hierbei, dass lediglich Traffic, der das entsprechende Netzsegment beziehungsweise die entsprechende Netzinfrastruktur durchläuft, analysiert werden kann.

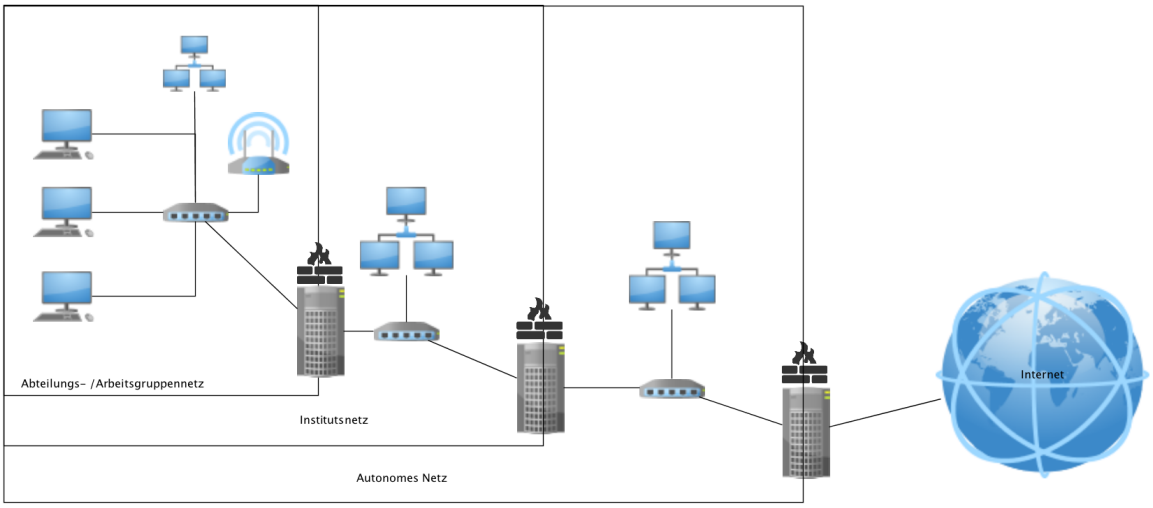


Abbildung 5.2.: Visualisierung von möglichen Erfassungsbereichen von Flow-Records [Jan]

Die Übertragung der durch die Netzkomponenten generierten Flow-Daten findet als unverschlüsselter UDP-Datenstrom an zentrale Flow-Sammler statt. Auf Grund der fehlenden Verschlüsselung der Flow-Daten wird für diese Übertragung ein eigenes durch ein V-LAN abgegrenztes Netz genutzt.

Der Flow-Sammler (oder auch Flow-Kollektor genannt) übernimmt nach dem Empfang der Flow-Records deren Speicherung in einem Binärdatenformat. Für diese Speicherung wird eine temporäre Datei genutzt, die in einem festen Zeitintervall abgeschlossen und mit dem Startzeitpunkt benannt wird. Im Anschluss wird eine neue temporäre Speicherdatei verwendet. Ab diesem Zeitpunkt kann die fertig geschriebene Binärdatei in weitere Formate exportiert beziehungsweise in eine Datenbank importiert werden.

Werden mehrere Flow-Kollektoren eingesetzt, so ist es notwendig, deren Daten an einem zentralen Server zusammenzuführen. Hierfür werden die Daten von dem zentralen Server aus periodisch über eine verschlüsselte Verbindung (z.B. SFTP) abgerufen und im Anschluss vom Kollektor entfernt. Diese abgerufenen Dateien werden durch Tools in eine sich in einem abgegrenzten Netz befindende und nicht von außen erreichbare Datenbank importiert. Nach dem Import werden diese auf dem zentralen Server liegenden Dateien zur Gewährleistung des Datenschutzes ebenfalls gelöscht, so dass die Daten nur noch in der Datenbank vorhanden sind

(vgl. Abbildung 5.3). Die Speicherung der Flow-Records in einer Datenbank ist notwendig, um die Geschwindigkeit der Auswertung zu optimieren sowie eine Durchsuchbarkeit der Datensätze zu ermöglichen. Des Weiteren können dadurch strukturierte Anfragen gestellt sowie einheitliche Schnittstellen genutzt werden, ohne hierbei ein eigenes Datenformat erstellen zu müssen.

Zur Speicherung der Daten gehört ebenfalls die Bereinigung der Datenbank von veralteten oder abgelaufenen Daten. Hierbei müssen mehrere Faktoren betrachtet werden.

Schützenswert beziehungsweise sensibel sind für diese Arbeit lediglich die über Personen gespeicherte Daten. Hierzu zählt insbesondere die IP-Adresse eines Nutzers, die in Flow-Records sowie weiteren Inhalten der Datenbank enthalten sein kann. Das BDSG (§3a BDSG – vgl. Abschnitt D.2.1) sowie das BayDSG (§15 ff BayDSG – vgl. Abschnitt D.3) fordern eine Datensparsamkeit sowie besondere Aufmerksamkeit beim Umgang mit personenbezogenen Daten sowie deren automatischer Verarbeitung. Aus diesem Grund sind verschiedene Methoden zur automatischen Löschung von Daten vorgesehen. Zum einen geschieht die Löschung der aufgezeichneten Flow-Record-Dateien, wie bereits erwähnt, unmittelbar nach deren Import in die entsprechende Datenbank, zum anderen werden die in der Datenbank gespeicherten personenbezogenen Daten automatisch nach einer festgelegten Speicherdauer von x Tagen entfernt. Die in der Asset-Datenbank erfassten Daten werden ebenfalls automatisch nach einer festgelegten Anzahl an Tagen, die von der Größe des Netzes abhängt sowie bei längerer Inaktivität des Hosts, entfernt. Für Server, die Dienste bereitstellen, also alle Hosts, die bei der Klassifikation als Server erkannt und getaggt wurden, gilt kein besonderer Schutz der erfassten und gespeicherten Daten. Aus diesem Grund werden über Server gespeicherte Daten erst entfernt, wenn der Server für einen vorher festgelegten Zeitraum inaktiv war beziehungsweise der Datenbankeintrag seit mehr als einer für den Anwendungsfall (abhängig von der Vergabe der IP-Adressen etc.) festgelegten Anzahl an Tagen als deaktiviert markiert ist.

An dieser Stelle ist es wichtig festzuhalten, dass die Möglichkeit besteht Anwenderdaten in den Serverdaten zu finden. Dies kann zum Beispiel bei VPN-Servern auftreten. Sobald die Wahrscheinlichkeit gegeben ist, dass Userdaten enthalten sind, müssen die Serverdaten partiell wie Anwenderdaten behandelt werden. In diesem Fall werden zwar die regulären Informationen über ein Asset gespeichert (zum Beispiel betriebene Dienste und IP-Adresse), Flow-Records jedoch nach Ablauf der Höchstspeicherdauer entfernt.

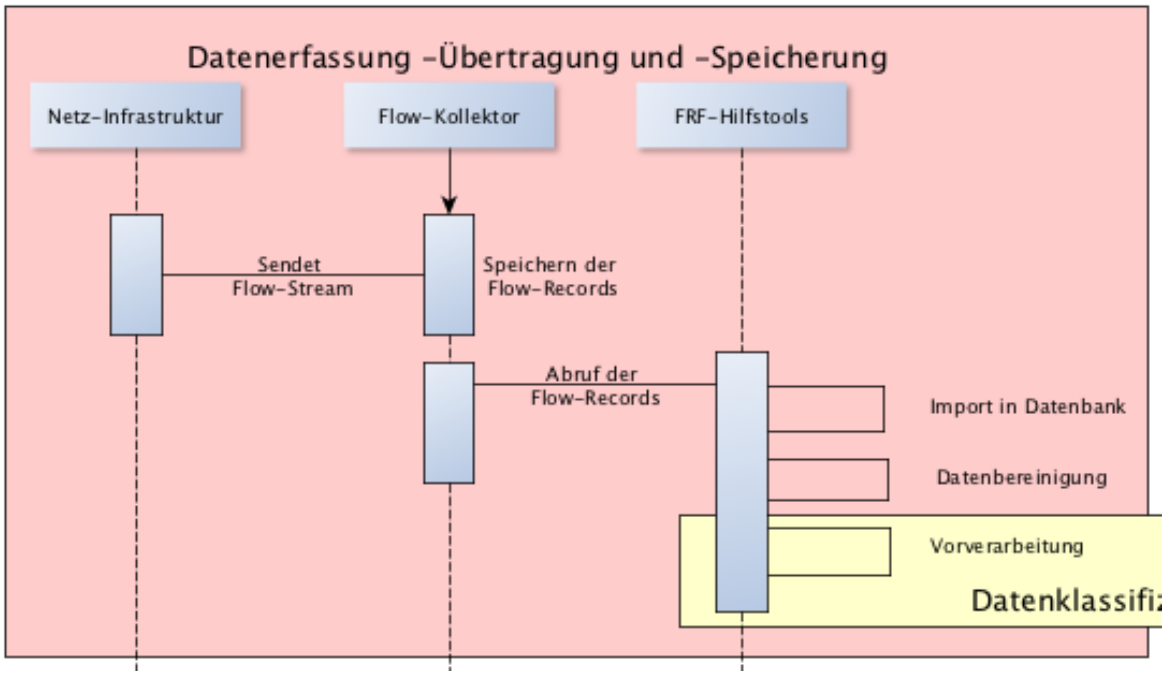


Abbildung 5.3.: Ablauf von Datenerfassung bis -Speicherung

Setzt man das oben beschriebene Verfahren (Datenerfassung, -Übertragung und -Speicherung) in einem großen Netz mit vielen Teilnehmern wie beispielsweise dem Münchner Wissenschaftsnetz ein, so ist es nötig, die Flow-Records an mehreren Punkten im Netz zu erfassen. Hierfür eignen sich sowohl die Gateways und Switches des Netzes, als auch die vorhandene WiFi-Infrastruktur. Um einen möglichen Datenverlust durch Netzprobleme zu vermeiden, sollten ebenfalls mehrere Flow-Kollektoren genutzt werden. Die gesammelten Daten müssen hierbei periodisch in die zentrale Datenbank eingespielt werden. Abbildung 5.4 verdeutlicht die vorhergehend genannten Erfassungsmöglichkeiten anschaulich und zeigt deren Hierarchie in großen Netzen auf.

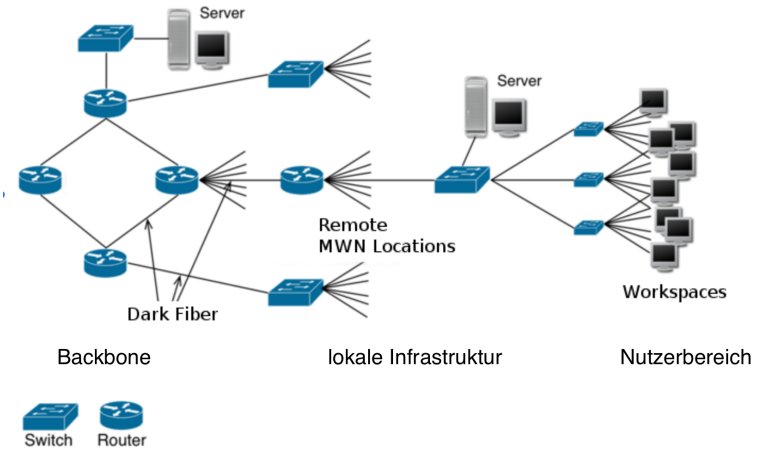


Abbildung 5.4.: Schematisierung der Struktur des MWN nach [HR15]

5.2. Datenklassifizierung

Im Anschluss an die Erfassung, Übertragung sowie die Speicherung der Flow-Records ist es notwendig, eine Vorverarbeitung dieser Flow-Records durchzuführen. Hierbei werden, wenn möglich, den Flow-Records auf Basis von einfachen Mustern Klassen zugewiesen. Zu diesen Klassen zählen unter anderem die Art des Hosts oder eines angebotenen Services. Beispiele für die hier genutzten Muster stellen Standardports oder Zugriffe von außen dar.

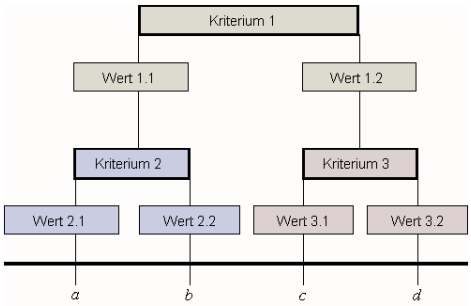


Abbildung 5.5.: Visualisierung von Klassifikation [Chr]

Um potentielle Ungenauigkeiten zu vermeiden, ist es an dieser Stelle notwendig, einige zentrale Begriffe für deren Verwendung zu definieren. Hierbei handelt es sich um die Begriffe Klassifizierung, Asset und Asset-Klassifizierung.

Definition 5.1

*Unter einer **Klassifizierung** wird das Zusammenfassen verschiedener Objekte in Klassen, welche nach bestimmten charakteristischen Merkmalen gebildet werden, verstanden. Hierbei ist es möglich, dass die klassenbildenden Merkmale bereits aus dem Ordnungsbegriff der Klassen erkennbar werden. Das Durchführen einer Klassifizierung erlaubt es, ein Ordnungssystem zu bilden. Um eine Eindeutigkeit der Zuordnung in die verschiedenen Klassen zu gewährleisten, kann es notwendig sein, Klassengrenzen zu definieren. Ferner ist es auch möglich, Klassen hierarchisch ineinander zu verschachteln, um unter anderem eine Suchstruktur aufzubauen (vgl. Abbildung 5.5).*

Abbildung 5.6.: Visualisierung IT-Assets [Jan]

Zur Wiederholung wird die in Kapitel 2 angerissene Erläuterung des IT-Assets nachfolgend definiert:

Definition 5.2

*Unter dem Begriff **Asset** wird im Rahmen dieser Arbeit jedwede IT-Ressource verstanden. Im Gegensatz zur Definition aus ITIL werden im Rahmen dieser Arbeit lediglich die Asset-Typen Anwendung und Infrastruktur betrachtet [AXE11] (vgl. Abbildung 5.6).*

Somit stellt ein Asset die Kombination aus Rechner-Infrastruktur sowie dort eingesetzter Anwendungen und Betriebssysteme dar.

Definition 5.3

*Unter einer **Asset-Klassifizierung** wird im Rahmen dieser Arbeit das Klassifizieren der detektierten IT-Assets, also das Kategorisieren und Gruppieren dieser, verstanden. Die Asset-Klassifizierung ist abhängig von der Nutzungsart der Rechner, den betriebenen Diensten sowie der eingesetzten Software.*

Entsprechend lässt sich die Vorverarbeitung in drei Hauptschritte, nämlich die Host-Klassifizierung, die Service-Klassifizierung sowie die Software-Klassifizierung unterteilen. Festzuhalten ist, dass die Klassifizierung im Rahmen der Vorverarbeitung lediglich einen Indikator, nicht jedoch ein finales Resultat liefert.

5.2.1. Hostklassifizierung

Die erste notwendige Klassifizierung der detektierten Assets findet nach Art des vorliegenden Hosts statt. Hierbei ist es neben der Unterscheidung nach Server und Anwendergerät auch möglich nach weiteren Gerätearten wie beispielsweise Mobiltelefonen zu differenzieren. Eine Trennung zwischen persönlich genutzten Geräten und Servern wird notwendig, um datenschutzrelevante Informationen über Nutzer nach Erreichen der Höchstspeicherdauer automatisch löschen zu können. Die Löschung der die Server betreffenden Datensätze ist nicht notwendig. Kann ein Host nicht eindeutig als Server klassifiziert werden, so wird dieser immer als Nicht-Server eingestuft, um die Regelungen des Datenschutzes einzuhalten.

Im Rahmen dieser Arbeit findet die Betriebssystemklassifizierung ebenfalls während der Host-Klassifizierung statt. Hierfür werden Cluster der verschiedenen Betriebssystemgruppen sowie Betriebssysteme erstellt. Diese Cluster können als Baumstruktur verstanden werden, wobei die Wurzel der Betriebssystembäume die Hersteller wie beispielsweise Microsoft, Apple oder Canonical sein können. Gefolgt vom Hersteller stellen die Betriebssystemversionen sowie der Patchstand weitere Knoten des Baums dar.

Sofern keine vollständige oder nur eine partielle Klassifizierung erfolgen kann, werden nur die erkannten Klassen zugeordnet.

An dieser Stelle ist festzuhalten, dass der Prozess der Klassifizierung nach der vorhergehend beschriebenen Vorklassifizierung nicht abgeschlossen ist, sondern durch weitere Schritte ergänzt oder geändert werden kann.

5.2.2. Dienstklassifizierung

Definition 5.4

*Unter einem **Dienst** ist ein für alle oder bestimmte Teilnehmer eines Netzes bereitgestellter Service zu verstehen. Ein derartiger Dienst ist unter der Adresse des Hosts mit einem oder mehreren spezifischen Ports zu erreichen. An einem Host kann unter jedem Port jeweils nur ein spezifischer Dienst bereitgestellt werden. Wichtig hierbei ist, dass mit einem Service nicht die installierte Software, sondern alleine der nach außen bereitgestellte Dienst zu verstehen ist.*

Eine Schwierigkeit bei der Erkennung von Diensten liegt daran, dass verschiedene Services durch unterschiedliche Software unter den gleichen Ports zur Verfügung gestellt werden können. Ein Beispiel für diese Problematik stellt der Betrieb eines Webservers dar. Für den Betrieb eines Webservers können unter anderem Internet Information Services (IIS), Apache oder NGINX mit den Standartports 80 sowie 443 genutzt werden, wobei diese Liste keinen Anspruch auf Vollständigkeit besitzt. Allein durch Betrachtung der anfallenden Verbindungen ist also lediglich feststellbar, dass der betriebene Service der Klasse Webserver zuzuordnen ist. Neben dieser Schwierigkeit ist es auch möglich, dass grundverschiedene Services standardmäßig die gleichen Ports nutzen beziehungsweise unter benutzerdefinierten und somit vorher nicht bekannten Ports betrieben werden.

Während der Service Klassifikation werden mehrdeutige Dienste/Services nach Art des Dienstes (SSH-Server, Webserver, etc.) gruppiert und erst in der nachfolgenden Flow-Record-Analyse hinsichtlich eingesetzter Software weiter untersucht. Des Weiteren werden die für einen Host festgestellten Serviceklassen in der Asset Datenbank mit diesem zu einem Host-Service-Tupel verbunden und im weiteren Verlauf genutzt. Ist keine Klassifikation möglich, so wird einem Dienst keine Klasse zugeordnet.

5.2.3. Softwareklassifizierung

Neben den Services, die die nach außen angebotenen Dienste darstellen, ist es notwendig, auch die auf Hosts installierte Software sowie den zugehörigen Patchstand zu klassifizieren. Zur Abgrenzung von der vorherigen Definition der Services ist es notwendig zu Beginn dieses Abschnitts den Begriff der Software zu definieren.

Definition 5.5

*Im Rahmen dieser Arbeit werden unter dem Begriff **Software** die auf einem Rechner installierten und betriebenen Anwendungen verstanden. Im Gegensatz zu einem Service besitzt Software eine eindeutige Bezeichnung. Für den Betrieb eines Dienstes wird eine entsprechende Software eingesetzt.*

Bei der Klassifizierung eingesetzter Software wird eine Baumstruktur mit der Hierarchie Hersteller, Bezeichnung und Version eingesetzt. Ist es nicht möglich alle Informationen in der Klassifizierung zu eruieren, so werden lediglich die erkannten Informationen erfasst.

5.3. Datenauswertung

Die Auswertung der in die Datenbank importierten und vorverarbeiteten Flow-Records basiert auf zwei wesentlichen Schritten, nämlich zunächst dem Training des FRF-Tools und im Anschluss der Auswertung der Daten mit dessen Hilfe.

Im ersten Schritt ist es notwendig, ein geeignetes Set an Trainingsdaten bereitzustellen, um das im FRF-Tool genutzte heuristische Verfahren anzulernen. Diese Trainingsdaten werden in einem Labornetz, welches vollständig kontrolliert wird, erfasst. Somit besteht ein vollständiges Wissen über alle vorhandenen Assets. Hierbei wird im Trainingsnetz für jedes zu erkennende System ein geeigneter Host betrieben. Im zweiten Schritt, im Anschluss an das Training, werden die Flow-Records durch eine heuristische Analyse ausgewertet.

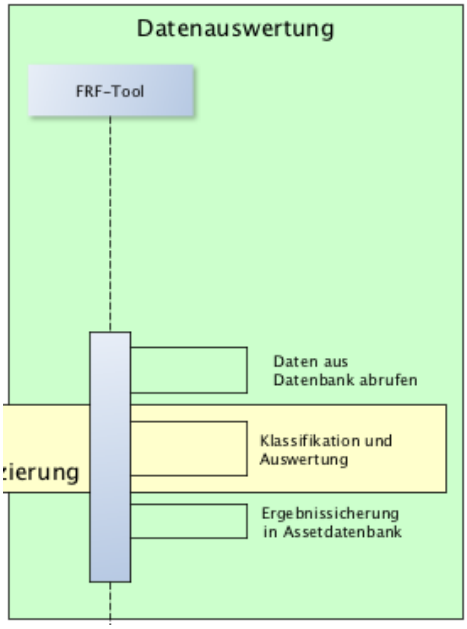


Abbildung 5.7.: Ablauf der Datenauswertung

Da es nur möglich ist, bereits bekannte Systeme zu erkennen, ist es neben der Auswahl der geeigneten Trainingsdaten auch notwendig, ein Verfahren zur Erlangung beziehungsweise Generierung dieser Daten zu beschreiben. Dieses Verfahren zur Datengenerierung, kurz Samplegenerator genannt, nutzt ein dediziertes Netz, in dem betriebssystemspezifische Anfragen wie Updates, Service-Kommunikation oder andere für die eingesetzten Betriebssysteme definierten Anfragen periodisch durchgeführt werden. Diese werden unmittelbar im angrenzenden Gateway erfasst und mit den eingesetzten Betriebssystemen sowie angebotenen Services und genutzter Software getaggt.

Für eine möglichst genaue Erkennung von Daten, also die Erkennung von Ähnlichkeiten mit bereits bekannten und für das Training genutzten Daten, sind qualitativ hochwertige Trainingsdaten notwendig. Hierbei ist unter der Qualität der Trainingsdaten eine möglichst hohe Eindeutigkeit zu verstehen. Dies bedeutet, dass die für das Training eingesetzten Datenklassen zwar das volle Spektrum an Betriebssystemen und Software abbilden, jedoch so wenig Schnittmengen zwischen unterschiedlichen Systemen wie möglich vorhanden sind. Zudem ist es notwendig, für jedes zu detektierende Betriebssystem beziehungsweise für jeden zu detektierende Service sowie die zu detektierende Software ausreichend Beispieldaten vorzuhalten. Damit geht einher, dass es nur möglich ist, vorher trainierte und somit erlernte Assets erkennen zu können.

Betrachtet man den Aufbau der in die Datenbank importierten Flow-Records, insbesondere hierbei die dort vorhandenen Felder, so ist es ebenfalls notwendig, die Auswahl der Datenfelder auf die Aussagekräftigsten einzuschränken. Die in den importierten Datensätzen enthaltenen Felder werden in Tabelle 5.1 genannt.

Werden in der Datenbank Felder nicht befüllt, weil beispielsweise der eingesetzte Flow-Recorder hierfür keine Daten sammelt, so ist es sinnvoll, diese nicht für das Training einzusetzen.

Name	Beschreibung
unix_secs	Anzahl der Sekunden seit 00:00 UTC 1970
unix_nsecs	Restliche Nanosekunden seit 00:00 UTC 1970
sysuptime	Zeit in Millisekunden, seit der Export-Host gebootet wurde
exaddr	IP-Adresse des Export-Hosts
dpkts	Anzahl der Pakete im Flow
doctets	Anzahl der Layer 3 Bytes innerhalb der Pakete des Flows
first	Uptime bei Start des Flows
last	Uptime bei Ende des Flows
engine_type	Art der switching engine: RP = 0, VIP/Linecard = 1
engine_id	ID der switching engine
srcaddr	Quell IP Adresse
dstaddr	Ziel IP Adresse
nexthop	IP Adresse des nächsten Routers
input	SNMP index des Input Interfaces
output	SNMP index des Output Interfaces
srcport	TCP/UDP Quellsystem Port
dstport	TCP/UDP Zielsystem Port
prot	IP Protokoll Typ (z.B. TCP = 6; UDP = 17)
tos	IP Typ des Services (ToS)
tcp_flags	TCP Flags
src_mask	Quell Adressmaske in Bits
dst_mask	Ziel Adressmaske in Bits
src_as	Autonomes-System-Nummer der Quelle, entweder Herkunft oder Peer
dst_as	Autonomes-System-Nummer des Ziels, entweder Herkunft oder Peer

Tabelle 5.1.: In Flow-Records enthaltene Datenfelder

Auch die Daten über den Flow-Exporter wie beispielsweise *sysuptime* oder *exaddr* sollen nicht genutzt werden, da diese keine Informationen über den zu analysierenden Host liefern können. Informationen über das Netzsegment, wozu der nächste Router, Netzmasken oder auch die IP-Adresse des Hosts zählen, eignen sich ebenfalls nicht für das Training der Erkennungsmetrik, da diese Daten zu falschen Korrelationen führen können.

Neben der Definition der für eine Wiedererkennung irrelevanten Daten ist es auch notwendig, die relevantesten Daten aufzuzeigen. Für eine hohe Relevanz ist es bei den Trainingsdatensätzen notwendig, für Betriebssysteme spezifische und einmalige Inhalte zu nutzen sowie bei der Auswahl der Felder möglichst konkrete Korrelationen zwischen dem Betriebssystem und dem Datensatz herzustellen (zum Beispiel Ziel-IP+Port, Uhrzeit von Updates).

Da eine Vielzahl an möglichen Kombinationen an Feldern existiert, ist es sinnvoll ein geeignetes Verfahren zur Bestimmung der relevanten Felder zu nutzen. Hierfür ist es möglich, eine sogenannte Kreuzvalidierung einzusetzen.

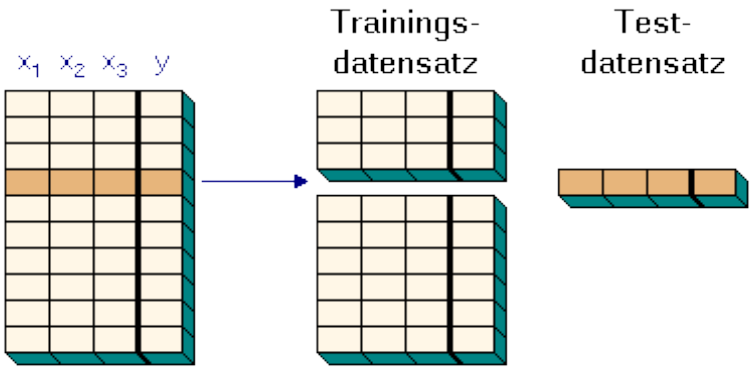


Abbildung 5.8.: Visualisierung von Kreuzvalidierung [Han]

Bei der Kreuzvalidierung werden die Trainingsdaten in einen größeren und einen kleineren Teil aufgeteilt, wobei der größere Teil für das Training verwendet wird und der kleinere Teil dazu dient, das trainierte Modell zu verifizieren. Maßgeblich hierbei ist die erreichte Erkennungsrate, also wie viele der Datensätze korrekt wiedererkannt werden können. Durch Variation der für das Training genutzten Datenfelder ist es möglich, ein optimales Set an Datenfeldern zu erreichen [Han].

Neben der Auswahl der Datenfelder eignet sich die Kreuzvalidierung auch, um ein möglichst optimales Set an Trainingsdaten zu erhalten. Hierfür nutzt man ein konstantes Set aus Testdaten, um mit diesem eine Auswahl an Trainingsdaten mit möglichst hoher Erkennungsrate zu erlangen.

Im Anschluss an die Definition der genutzten Datensätze sowie der Erstellung von Trainingsdaten wird mittels eines heuristischen Verfahren untersucht, um welches System es sich am wahrscheinlichsten handelt. Die heuristische Untersuchung gliedert sich in drei Schritte. Zuerst wird das vorhandene Betriebssystem detektiert, im Anschluss hieran werden die angebotenen Services untersucht und abschließend die eingesetzte Software analysiert. Die so gewonnenen Informationen werden anschließend mit den bereits in der Asset-Datenbank vorhandenen Informationen abgeglichen. Sofern die erfassten Informationen zu den bereits vorhandenen Informationen passen, wird der Eintrag in der Datenbank ergänzt, falls nicht wird der Eintrag durch die neueren Informationen ersetzt. Neben den Informationen über eingesetzte Software, angebotene Dienste sowie genutztes Betriebssystem enthält die Asset-Datenbank auch die Adresse des Hosts sowie den Zeitpunkt, zu dem der Host zuletzt im Netz detektiert wurde. Ist die Klassifizierung nicht eindeutig möglich, so wird die wahrscheinlichste Klasse zugewiesen.

5.3.1. Heuristische Verfahren

Für die Auswertung durch heuristische Verfahren ist es möglich, verschiedene Verfahrensarten zu nutzen. Nachfolgend werden kurz die Familien der Bayes-Klassifizierung, Entscheidungsbäume sowie neuronale Netze beschrieben. Da die für diese Arbeit genutzten Verfahren offen zugänglich sind, wird auf die Erläuterung der individuellen Algorithmen verzichtet.

Bayes-Klassifikation
Die erste Möglichkeit stellen hierbei klassische Klassifikatoren dar, zu denen beispielsweise der Bayes-Klassifikator zählt. Bei dieser Art der Klassifikation wird jedem Objekt, in diesem Fall jedem Flow-Record, die Klasse zugeordnet, zu dem dieser mit der größten Wahrscheinlichkeit gehört. Im Allgemeinen lässt sich über den Bayes-Klassifikator, für den es verschiedene Implementierungen gibt, festhalten, dass dieser eine mathematische Funktion nutzt, welche jedem Punkt des Merkmalsraums eine Klasse zuordnet. Bei Nutzung des Bayes-Klassifikators werden jeder Klassifizierung Kosten (durch die sogenannte Risikofunktion) zugewiesen, wobei eine Minimierung der Kosten angestrebt wird. Festzuhalten ist jedoch, dass beim Bayes-Klassifikator alle Merkmale gleich gewichtet sind und vorausgesetzt wird, dass alle stochastisch voneinander unabhängig sind.

Zentraler Vorteil der Bayes-Klassifikation sind eine hohe Geschwindigkeit sowie Genauigkeit auf großen Datenmengen. Der zentrale Nachteil zeigt sich jedoch, falls die Annahme der Unabhängigkeit der Attribute nicht zutrifft, da in diesem Fall die Ergebnisse ungenau werden. [Dir02]

Entscheidungsbäume
Eine weitere Untersuchungsmöglichkeit stellt die Klassifikation mit Hilfe von Entscheidungsbäumen (aufbauend auf den ID3-Algorithmus) dar. Hierbei werden von der Wurzel ausgehend die Attribute aller Knoten geprüft und abhängig von der Ausprägung den entsprechenden Verzweigungen gefolgt. Die Klassifikation ist bei Ankunft in einem Blattknoten, dessen Beschriftung die Klasse angibt, abgeschlossen.

Für die Konstruktion von Entscheidungsbäumen wird ein Divide-and-Conquer-Verfahren auf Basis der Trainingsdaten durchgeführt. So wird in jedem Knoten mit einer informationstheoretischen Kennzahl entschieden, anhand welchen Attributs die nächste Verzweigung geschehen soll, wobei für jede existierende Ausprägung eine Verzweigung gebildet und der Algorithmus mit den zur gleichen Klasse gehörenden Trainingsobjekten rekursiv weitergeführt wird. Sobald an einer Verzweigung alle Trainingsobjekte zur gleichen Klasse gehören, wird dort ein Blattknoten dieser Klasse gebildet. Sofern alle Attribute genutzt wurden, jedoch kein eindeutiges Ergebnis besteht, wird ein Blattknoten für die häufigste Klasse gebildet.

Als wesentliche Vorteile von Entscheidungsbäumen lassen sich die einfache und schnelle Umsetzbarkeit, sowie die Optimierbarkeit durch gezielte Positionierung innerhalb des Entscheidungsbaums nennen. Negativ ist jedoch anzumerken, dass bei Entscheidungsbäumen abhängig von der eingesetzten Implementierung die Trainingsdaten oftmals vollständig im Hauptspeicher gehalten werden müssen. [Dir02]

Neuronale Netze

Unter neuronalen Netzen versteht man ein Netz aus Knoten, den sogenannten Neuronen, die miteinander verbunden sind und sich gegenseitig aktivieren. In der gebräuchlichsten Form (fully-connected, feedforward, multilayer perceptrons) besteht ein neuronales Netz aus mehreren Schichten. Diese Schichten teilen sich in Eingabeschicht, mehrere verborgene Schichten sowie die Ausgabeschicht auf, wobei jedes Neuron mit allen Neuronen der nachfolgenden Schicht verbunden ist. Zu Beginn besitzen die Kanten ein zufälliges Gewicht.

Während der Lernphase werden die Merkmale des Trainingsobjekts als numerische Daten an die entsprechenden Neuronen der Eingabeschicht übergeben und von dort aus gewichtet an die Neuronen der ersten verborgenen Schicht weitergeleitet. Im Anschluss bildet jedes Neuron die gewichtete Summe über die erhaltenen Eingabedaten, übergibt dieses an eine Aktivierungsfunktion und leitet es im Anschluss an die Neuronen der nächsten Schicht weiter. An Hand des Ergebnisses, das durch die Neuronen in der Ausgabeschicht geliefert wird, ist es möglich, das Klassifikationsergebnis abzulesen.

In der Regel wird für jede Klasse ein Ausgabeneuron genutzt, das wenn die zugehörige Klasse als Ergebnis herauskommen soll, als einziges aktiviert wird. Das Lernen erfolgt hier nach dem sogenannten Backpropagation-Ansatz, also dem Vergleich des erwünschten Ergebnisses mit der Ausgabe des Netzes, wobei die Differenz in umgekehrter Richtung an das Netz zurückgegeben wird. Das Lernen sorgt hierbei für eine langsame Anpassung der Gewichte, so dass die Klassifikation immer besser wird. Die Lernphase wird beendet, sobald sich kaum noch Veränderungen ergeben, die Klassifikation gut genug erscheint oder eine festgelegte Zeit abgelaufen ist.

Die Anwendung der gelernten Klassifikationsregeln geschieht im Anschluss sehr schnell.

Als Vorteil der neuronalen Netze ist festzuhalten, dass diese mit nicht erlernten Merkmalskombinationen sehr gut umgehen können.

Neuronale Netze gehen jedoch auch mit verschiedenen Nachteilen einher. Die Interpretation der Gewichte ist nur sehr schwer möglich, so dass die Klassifikationsergebnisse schwierig zu erklären sind. Insbesondere die Trainingsphase dauert sehr lange, wenn eine hohe Anzahl an Attributen vorliegt. Im schlechtesten Fall ist es möglich, dass keine Lösung gefunden wird. Entsprechend ist es notwendig, die genutzten Daten sorgfältig vorzubereiten. [Dir02]

5.3.2. Auswahl des heuristischen Verfahrens

Basierend auf den vorhergehend vorgestellten heuristischen Verfahren ist festzuhalten, dass diese unabhängig voneinander einsetzbar sind. Generell ist folglich jedes der beschriebenen Verfahren bei der Datenanalyse einsetzbar, jedoch wird nur ein Verfahren, nämlich das mit den besten Klassifizierungsergebnissen benötigt.

Jedes der verschiedenen heuristischen Verfahren geht mit individuellen Vor- und Nachteilen einher. Da die Datenanalyse eine schnelle Verarbeitung benötigt und in großen Netzen mit vielen Teilnehmern einsetzbar sein soll, bieten sich Verfahren aus der Familie der Bayes-Klassifikatoren sowie der Entscheidungsbäume an. Neuronale Netze sind auf Grund der oft hohen Dauer des Modelltrainings weniger gut für die Datenanalyse geeignet.

5.4. Zusammenfassung

Die in den vorhergehenden Abschnitten beschriebene Datenerfassung, -Übertragung, -Speicherung, -Klassifizierung und -Auswertung ist nachfolgend in Abbildung 5.9 schematisch visualisiert.

In dieser Abbildung wird der Datenverkehr eines Netzsegments mit dem Internet dargestellt. Die Netzkomponenten erfassen hier die Flow-Records und senden diese an den ihnen zugewiesenen Flow-Kollektor. Die so durch den Flow-Kollektor empfangenen Flow-Records werden zentral gespeichert und durch ein heuristisches Verfahren des FRF-Tools ausgewertet. Die Ergebnisse der Auswertung werden in die Asset-Datenbank gespeichert.

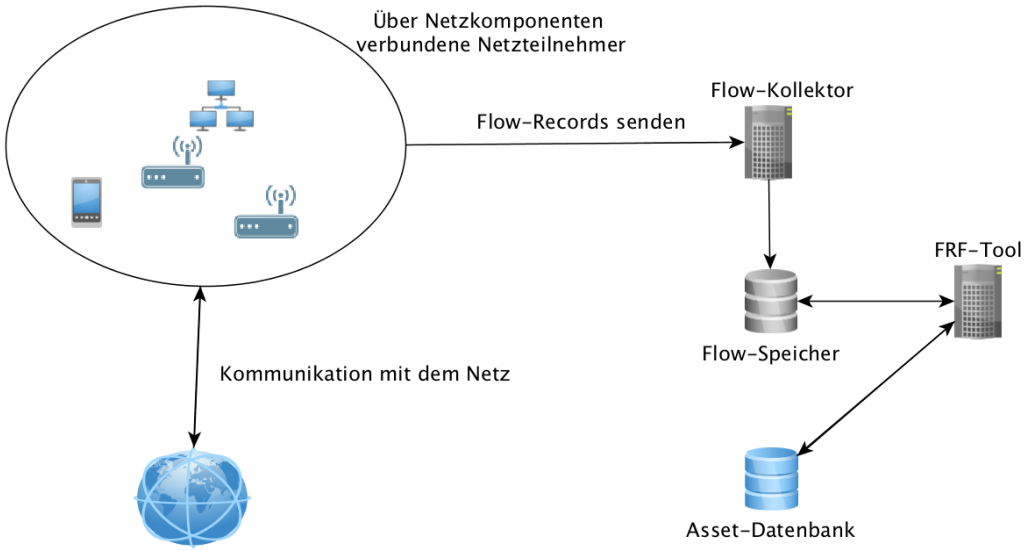


Abbildung 5.9.: Sammlung und Verarbeitung von Flow-Records

6. Konfiguration und Implementierung

Dieses Kapitel beschreibt die Konfiguration und Implementierung des in Kapitel 5 beschriebenen Konzepts in einem Proof of Concept. Hierfür werden zu Beginn die Grundlagen der Implementierung erläutert und anschließend der Aufbau des Labornetzes beschrieben. Darauf folgen die Beschreibung der Erlangung der Trainingsdaten sowie die im Proof of Concept genutzte Datenerfassung, - Übertragung und Speicherung der Flow-Records des Labornetzes. Abschließend werden die Datenklassifizierung sowie die Datenauswertung, in welcher auch kurz auf die heuristischen Verfahren eingegangen wird, und die Ergebnissicherung beschrieben. Im Anschluss folgt eine kurze Zusammenfassung des Kapitels.

6.1. Grundlagen

Im Rahmen dieses Abschnitts werden die wesentlichen Grundlagen, die für die Implementierung genutzt werden, beschrieben.

Um das in Kapitel 5 beschriebene Konzept umzusetzen, wird auf Grund der reduzierten Komplexität eine virtuelle Infrastruktur eingesetzt. Hierfür werden sowohl die beteiligten Rechner als auch die Netzinfrastruktur mit Hilfe von VMware Esxi 6[1] virtualisiert. VMware Esxi wurde auf Grund der Verbreitung dieses Tools sowie der einfachen Handhabung und Reproduzierbarkeit ausgewählt.

Als Betriebssystem zentraler Netzkomponenten wie beispielsweise Routern wird Debian Jessy eingesetzt. Die Konfiguration der eingesetzten Systeme erfolgt mit Hilfe des Automatisierungstools Ansible[2]. Die verwendeten Konfigurationsdateien sind in Anhang A beigelegt.

Seitens der Hardware bestehen keine speziellen Anforderungen. Wichtig ist lediglich eine ausreichende Rechenleistung für mehrere virtualisierte Maschinen sowie ausreichend Arbeitsspeicher und eine Verbindung mit dem Internet. Für den Proof of Concept wird ein in einem Rechenzentrum betriebener Server mit folgenden ausreichenden Kerndaten genutzt:

CPU	Intel Xeon E3-1245
RAM	4x RAM 8192 MB DDR3 ECC
Festplatte	2x HDD 1,5 TB SATA
Netzwerkinterface	Intel 82574L

Tabelle 6.1.: Serverhardware

Auf Grund seiner Verbreitung, der möglichen Anbindung verschiedener externer Datenbanksysteme sowie des Funktionsumfangs wird das Datamining Tool Weka[3] für die heuristische Untersuchung eingesetzt. Für die Nutzung von Weka mit einer MySQL-Datenbank ist es

[1] `http://www.vmware.com/products/vsphere-hypervisor.html`
[2] `https://www.ansible.com/`
[3] `www.cs.waikato.ac.nz/ml/weka/`

bei der hier verwendeten Version notwendig, Weka für die eingesetzten Datentypen vorzubereiten. Hierfür wird die Datei `weka/experiment/DatabaseUtils.props` um die benötigten Einträge für die in MySQL genutzten Formate, zum Beispiel
`INT_UNSIGNED=5`
ergänzt. Im Anschluss an diese Anpassung ist es möglich, die zu analysierenden Daten direkt aus der Datenbank auszulesen und zu analysieren.

6.2. Labornetz

Für die Implementierung sowie die Evaluierung des FRF-Tools ist es notwendig, ein eigenes, abgeschottetes und vollständig kontrolliertes Netz zu betreiben. Dieses Netz wird im Folgenden als Labornetz bezeichnet. Das wesentliche Merkmal dieses Labornetzes stellt hierbei das vollständige Wissen über alle Netzkomponenten sowie Netzteilnehmer dar.

Bei der Umsetzung des Labornetzes ist darauf zu achten, dass nicht nur logisch sondern auch physikalisch eine Abgrenzung erfolgt, damit keine eventuell störende Kommunikation vorhanden sein kann. Für die Kommunikation mit dem Internet, die unter anderem für das Abrufen von Updates notwendig ist, wird ein Gateway (Router) genutzt, welches gleichzeitig auch für die Network Address Translation (NAT) verantwortlich ist. Neben der Aufgabe der Vermittlung ist im Proof of Concept das Gateway auch für die Generierung sowie Erfassung und Speicherung der Flow-Records verantwortlich.

Um kontrolliert die Kommunikation mit betriebenen oder externen Diensten beziehungsweise Servern nachstellen zu können, wird neben diesem Teilnetz (Trainingsnetz), an welchem die untersuchten Systeme angeschlossen sind, ein weiteres Dienstnetz betrieben, welches durch den zentralen Router mit dem Labornetz verbunden ist. Ferner nutzen die so entstandenen Netze auch verschiedene Subnetze. Der Aufbau und die Struktur der Teststellung sind in Abbildung 6.1 visualisiert.

6.3. Generierung, Speicherung und Verarbeitung von Trainingsdaten

Wie in Kapitel 5 beschrieben kann das FRF-Tool lediglich bekannte Muster erkennen beziehungsweise Ähnlichkeiten zu diesen Mustern feststellen. Aus diesem Grund ist es notwendig, ein geeignetes Verfahren zur Generierung von Lerndaten zu erstellen. Dieses Verfahren wird im weiteren Verlauf Sample Generator genannt.

Das zentrale Kernelement des Sample Generators stellt das Labornetz dar, da die für das Training erfassten Daten in diesem gesammelt und aufgezeichnet werden. Einige der obersten Ziele bei der Generierung von Sample-Daten stellen die Reproduzierbarkeit der Daten sowie die hohe Qualität (Eindeutigkeit) dieser dar. Zur Erlangung dieser Ziele wird bei der Erfassung der Trainingsdaten ein Abbild der virtuellen Maschine angefertigt. Mit Hilfe dieses Abbilds ist es möglich, mehrfach zu dem bekannten Ausgangszustand zurückzukehren und Aktionen wie beispielsweise Updates oder Programminstallationen entsprechend häufig durchzuführen.

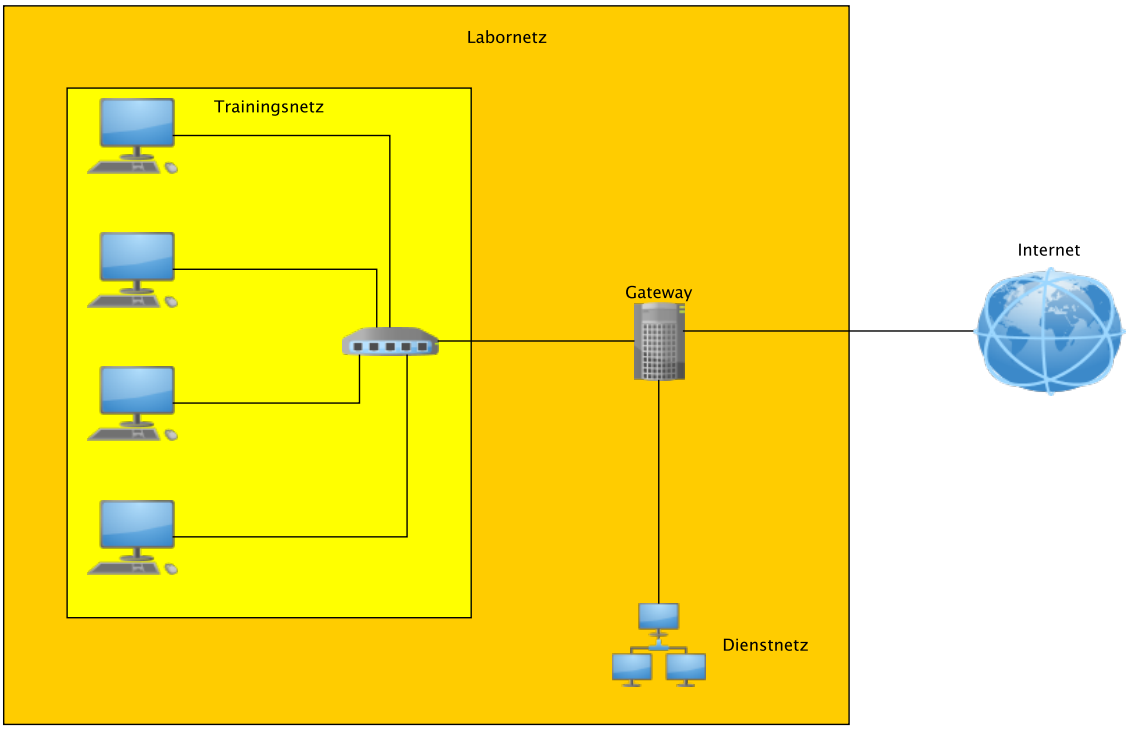


Abbildung 6.1.: Schematische Skizzierung der Teststellung

Die Erfassung der Trainingsdaten geschieht durch die Aufzeichnung der Hintergrundaktivitäten der einzelnen Betriebssysteme sowie durch Aufzeichnung der Suche nach verfügbaren Updates und deren Installation. Hierbei findet die Aufzeichnung, um ausreichend viele Daten über die Hintergrundkommunikation von Betriebssystemen zu erlangen, über einen Zeitraum von 10 Tagen statt.

Im Anschluss an die Erfassung der Trainingsdaten werden diese in der zentralen Datenbank gespeichert und um die entsprechend vorhandenen Informationen über Betriebssystem, eingesetzte Dienste sowie installierte Software angereichert. Diese Zuordnung geschieht im Proof of Concept durch fest codierte MySQL Anfragen und wäre als Ausblick in späteren Versionen zu automatisieren. Abbildung 6.2 visualisiert den Aufbau der genutzten Datenbank.

Diese Datenbank besitzt für eingehende Flow-Records die Tabelle `flowinput`, in der alle in Flow-Records enthaltenen Datenfelder vorhanden sind. Zusätzlich wird eine Spalte für Anmerkungen, welche für das Tagging nutzbar ist, hinzugefügt. Für die Speicherung der Trainingsdaten wird die Tabelle `trainingdata` genutzt. Zusätzlich zu den in Flow-Records enthaltenen Feldern sind hier die Spalten für Betriebssystem, Software, sowie Dienst in Form von Fremdschlüsseln ergänzt. Diese Fremdschlüssel verweisen auf die für die Asset Datenbank genutzten Tabellen `os`, `service` und `software`. Diese Tabellen besitzen für jede zu erkennende Information einen Eintrag (beispielsweise Windows 7 oder Apache). Neben der Verwendung in der Trainingsdatenbank werden diese Tabellen ebenfalls für die Asset-Datenbank genutzt, um Hosts mit Hilfe der Tabellen `asset_os`, `asset_service` und `asset_software`

Informationen zuzuweisen. Hierfür werden in diesen Tabellen Fremdschlüssel der Tabelle `assets`, die die Adresse sowie weitere Metainformationen über den Host beinhaltet sowie der Tabellen für Betriebssystem, Software und Services genutzt.

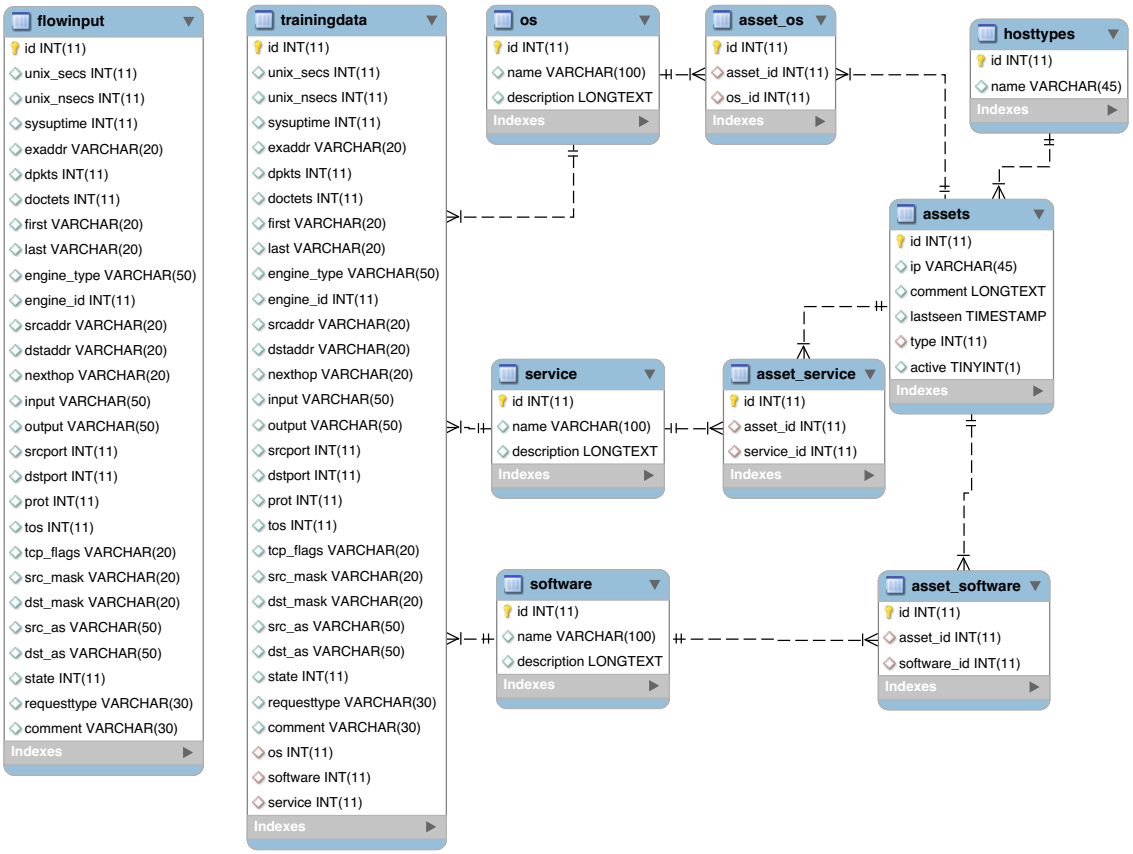


Abbildung 6.2.: Datenbanklayout des FRF-Tools

Um ein möglichst breites Spektrum an Trainingsdaten generieren zu können, werden mehrere Betriebssystemversionen sowie verschiedene Anwendungen und Dienste bereitgestellt. Die im Proof of Concept genutzten Betriebssysteme sind in Tabelle 6.2 dargestellt.

Windows	7
Windows	10
Debian	7
Debian	8
Ubuntu	14.04 Desktop
Ubuntu	14.04 Server
Ubuntu	16.04 Server
OpenSuse	12.2
Mac OS X	10.12

Tabelle 6.2.: Eingesetzte Betriebssysteme

Die Nutzung von iOS oder Android auf der virtualisierten Hardware ist auf Grund von herstellerseitigen und/oder technischen Einschränkungen leider nicht möglich, wäre in einem größeren Versuchsaufbau mit dedizierter Hardware jedoch analog zu den genannten Systemen möglich.

Hinsichtlich der Dienste werden die Webserveranwendung Apache 2, MySQL als Datenbankserver sowie Open-SSH – jeweils in der aktuellen Version – auf den Unix-basierten Betriebssystemen eingesetzt. Zusätzlich wird, sofern eine graphische Benutzeroberfläche vorhanden ist, TeamViewer als Remote-Desktop-Anwendung eingesetzt. Als installierte Anwendungen werden das Java Runtime Environment, Dropbox, Open Office und Skype verwendet.

6.4. Datenerfassung, -Übertragung und -Speicherung

Im Proof of Concept übernimmt das zentrale Gateway die Aufgabe der Generierung der Flow-Records. Zudem dient der Router auch als Flow-Kollektor, an den die Flow-Records gesendet werden. Die empfangenen Flow-Records werden bis zur weiteren Verarbeitung in einem Binärformat auf dem Dateisystem gespeichert.

Da die Datenbank, in welche die Flow-Records importiert werden, auf dem selben Rechner liegt, ist keine Datenübertragung über das Netzwerk notwendig und somit ein direkter Import in die Datenbank möglich. Der Import der gesammelten Flow-Records geschieht hierbei über einen Cronjob, der alle 4 Minuten das Importskript (vgl. Listing A.2) aufruft. Die Datenbank beziehungsweise die für den Import genutzte Tabelle enthält die in der Version 5 von NetFlows[4] [5] möglichen Felder (vgl. Abbildung 6.2).

Hierbei wird das Verzeichnis `/flowstore/data/`, in dem die Flow-Records gespeichert werden, nach vorhandenen fertig geschriebenen Datensätzen durchsucht, diese mit dem Befehl

```
flow-export -f3 -u 'thesis:thesis:localhost:3306:flows:flowinput' < $i
```

in die Datenbank flows importiert und die Quelldatei im Anschluss mit dem Befehl

```
mv $i /flowstore/old/$(basename $i)
```

in den Ordner `/flowstore/old` verschoben. Zwar ist im produktiven Einsatz an Stelle des Verschiebens eine Löschung der gesammelten Daten vorgesehen, jedoch geschieht dies im Proof of Concept nicht, um die gesammelten Daten einer erneuten Verprobung unterziehen zu können.

Neben dem Import der Binärdaten in die Datenbank übernimmt das genutzte Bashskript, welches im Anhang (Listing A.2) beigelegt ist, gleichzeitig die Bereinigung der Datenbank sowie die Vorverarbeitung. Zum Einen werden Flow-Records, die aus Übertragungen außerhalb des gewünschten Netzbereichs stammen, direkt nach dem Import entfernt, da eine Filterung vor dem Import nicht möglich ist und zum Anderen werden alle Flow-Records, die älter als 7 Tage sind, entfernt. Des Weiteren werden die Inhalte der Asset Datenbank, die länger als 7 Tage nicht aktualisiert wurden, ebenfalls entfernt. Zu diesem Zeitpunkt werden die bekannten Informationen ebenfalls zu den Flow-Records hinzugefügt.

[4] `https://www.plixer.com/support/netflow_v5.html`

[5] `http://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html`

6.5. Datenenklassifizierung

Die im Proof of Concept eingesetzte Datenklassifizierung lässt sich in drei Bereiche einteilen. Diese Bereiche sind die Host-Klassifizierung, zu der ebenfalls die Betriebssystemklassifizierung zählt, die Dienstklassifizierung sowie die Softwareklassifizierung.

6.5.1. Host

Auf Grund der Anforderung, den Datenschutz für personenbezogene Daten einzuhalten, wurde in Kapitel 5 eine Trennung zwischen Servern und sonstigen Hosts beschrieben. Standardmäßig wird zunächst jeder detektierte Host so lange als sonstiger Hosts angesehen, bis er als Server detektiert wurde.

Das wesentliche Entscheidungsmerkmal, das hierbei für die Erkennung eines Servers genutzt wird, stellt die Anzahl der eingehenden Verbindungen auf bekannten Serverports wie beispielsweise Port 80 dar. Wenn ein Rechner innerhalb von einer Stunde den Schwellwert von 5 eingehenden Verbindungen überschreitet, wird dieser Rechner als Server klassifiziert. Bei der Wahl des Kriteriums ist festzuhalten, dass verschiedene Anwendungen selbst auch Ports nach außen anbieten, um Updates zu verteilen oder Verbindungen gemeinsam nutzen zu können. Dies kann zum Beispiel bei der Verteilung von Updates in Windows 10 oder bei der Lastverteilung von Skype auftreten.

Neben der Klassifizierung nach Art des Hosts findet während der Hostklassifizierung auch die Detektion des Betriebssystems statt. Hierfür werden die im Labornetz generierten Trainingsdaten genutzt, um verschiedene heuristische Verfahren zu trainieren. Die Auswertung der Daten wird ausführlich in Abschnitt 6.6 beschrieben.

6.5.2. Dienst

Die Klassifizierung von Diensten basiert zum Einen auf der Liste der bekannten Ports und zum Anderen auf dem Wissen über das zu erwartende Antwortverhalten. Ein Beispiel hierfür stellt der Betrieb eines FTP-Servers dar, bei dem die initiale Anfrage auf Port 21 geschieht. Zudem wird je nach Verfahren ein dedizierter Port (> 1023) für den Client ausgehandelt. Die Datenübertragung geschieht auf der Serverseite durch Port 20.

Nutzt man dieses Wissen, so können mit geringem Aufwand Regeln festgeschrieben werden, die die Klassifizierung nach Art des Service sowie eine Plausibilitätsprüfung dieser Klassifizierung erlauben. Für Dienste, die auf Ports außerhalb des Standardbereichs gestartet werden (zum Beispiel Gameserver), lässt sich ein heuristisches Verfahren einsetzen, wobei hierfür vorab entsprechende Trainingsdaten benötigt werden.

Im Rahmen des Proof of Concept findet die Dienst-Klassifizierung für die gängigsten Dienste basierend auf den bekannten Ports statt.

6.5.3. Software

Die Klassifizierung der eingesetzten Software geschieht analog zur Betriebssystemdetektion. Hierfür werden im Trainingsnetz Flow-Records der zu identifizierenden Software generiert und mit den entsprechenden Informationen versehen. Da eine aktive Beeinflussung des Netzes durch die Anforderungen an das FRF-Tool ausgeschlossen wird, ist es nicht möglich, durch gezieltes Probing weitere Informationen über die Software eines Hosts zu sammeln.

6.6. Datenauswertung und Ergebnissicherung

Der nachfolgende Abschnitt ist in zwei Bereiche aufgeteilt. Im ersten Bereich, der Datenauswertung, wird beschrieben, wie im Proof of Concept die Daten aufbereitet und hinsichtlich Betriebssystem sowie Software untersucht werden. Ebenfalls wird in diesem Bereich auf die heuristische Untersuchung eingegangen. Der zweite Abschnitt beschreibt die anschließende Sicherung der Ergebnisse in der genutzten Asset Datenbank.

6.6.1. Datenauswertung

Vorverarbeitung und Informationsauswahl

Nach Import der gesammelten Flow-Records in die zentrale Datenbank findet eine Vorverarbeitung statt. Hierbei werden irrelevante Informationen wie beispielsweise die Kommunikation mit dem Router sowie DHCP- oder DNS-Anfragen und -Antworten entfernt. Diese Aufbereitung geschieht im Proof of Concept durch eine View. Mit Hilfe dieser Evaluations-View wird auch die Übertragungsdauer aus der Start- und Endzeit berechnet, so dass das für die Analyse genutzte Result-Set folgende Datenfelder beinhaltet:

dpkts	Anzahl der Datenpakete
doctets	Größe der Übertratung
duration	Dauer der Übertragung
srcaddr	Quelladresse
dstaddr	Zieladresse
scrport	Quellport
dstport	Zielport
prot	Protokoll
tcp_flags	TCP Flags

Tabelle 6.3.: Felder der Evaluations-View

Neben der oben beschriebenen Aufbereitung durch die View werden bei den Trainingsdaten auch die lokalen IP-Adressen (in diesem Fall, 10.150.254.0/24) entfernt. Die so aufbereiteten Datensätze werden für das Training des heuristischen Verfahrens genutzt.

Betriebssystemdetektion
Eine manuelle Betrachtung der aufbereiteten Flow-Records zeigt hinsichtlich der Anzahl der entstandenen Flow-Records deutliche Unterschiede zwischen den Betriebssystemen. Obwohl auf allen Hosts automatische Updates aktiviert sind und auch mehrfach Updates abgerufen wurden, ist erkennbar, dass die Serverbetriebssysteme kaum Netzkommunikation verursachen. Dafür ist eine wesentlich aktivere Kommunikation der Desktopsysteme festzustellen, so dass für diese mehr Trainingsdaten bereit stehen (Siehe Tabelle 6.4).

Betriebssystem	Anzahl der Flow-Records	Anzahl der Hosts
debian 7	16	2
debian 8	35	2
Mac OS X Sierra	1771	1
Open Suse 12	4760	2
Ubuntu 14 Desktop	232	1
Ubuntu 14 Server	30	1
Ubuntu 16 Server	786	2
Windows 10	6258	2
Windows 7	1214	2

Tabelle 6.4.: Anzahl der Flow-Records

Um die erfassten Daten systematisch zu untersuchen, wird eine heuristische Analyse mit Hilfe des Dataminingtools Weka 3 [6] durchgeführt, da dieses eine Vielzahl an heuristischen Verfahren aus den in Abschnitt 5.3.1 beschriebenen Bereichen implementiert hat und die Anbindung an verschiedene Datenbanksysteme erlaubt.

Heuristische Untersuchung

Um im Rahmen des Proof of Concept die Aussagekraft der ausgewählten Datensätze untersuchen zu können, findet die Evaluation auf dem Trainingsset selbst statt. Hierbei wird durch eine Kreuzvalidierung mit 10 Faltungen jeweils ein Teil des Trainingssets für das Training genutzt und ein kleinerer Teil klassifiziert. Am Ende werden die Ergebnisse der Klassifizierung mit den tatsächlichen Klassen verglichen und die Erkennungs- sowie Fehlerrate berechnet.

Um den für das FRF-Tool optimalen Klassifikationsalgorithmus aus den in Abschnitt 5.3.1 beschriebenen Verfahrensarten zu finden, wird für die gängigsten Verfahren aus den beschriebenen Kategorien eine Kreuzvalidierung durchgeführt und das Verfahren mit den besten Erkennungsraten ausgewählt. Nachfolgend wird die Untersuchung des Trainigssets der Betriebssystemklassifikation genutzt. Analog zu diesem Datenset gilt die Beschreibung des Verfahrens ebenfalls für die Softwareklassifikation. Die Ergebnisse dieser Validierung mit Hilfe der Betriebssystem-Trainingsdaten sind im Anhang C beigelegt.

Aus dieser Validierung der verschiedenen Algorithmen wird eine breite Varianz bezüglich der Fehlerrate erkennbar. So haben der traditionelle Naive Bayes sowie der Naive Bayes Multinomial Text weniger als 50% korrekt klassifiziert, während der BayesNet über 95% korrekt klassifizieren konnte. Betrachtet man nun die Ergebnisse von Entscheidungsbäumen, so fällt auf, dass in jedem Fall mehr als 50% der Datensätze korrekt klassifiziert werden konnten und bei J48 mit über 94% die beste Erkennungsrate bei den Entscheidungsbäumen vorliegt. Durch Variation der verfügbaren Aufrufparameter war es nicht möglich, bessere Ergebnisse als mit Hilfe der Standardparameter zu erzielen, so dass für die Analysen lediglich die Standardparameter verwendet werden. Eine exemplarische Nutzung weiterer, nicht im Proof of Concept eingesetzter Verfahren, erlaubte ebenfalls keine Resultate mit einer Erkennungsrate von über 50%.

[6] `http://www.cs.waikato.ac.nz/ml/weka/`

Der Vergleich der Zeit, die benötigt wird, um das Datenmodell aufzubauen, zeigt jedoch eine Differenz von mehr als dem Faktor 10 zu Gunsten des BayesNet-Verfahrens. Im Gegensatz zu den oben angesprochenen Verfahren war es in einer Zeit von 5 Stunden nicht möglich ein neuronales Netz mit dem Multilayer Perceptron zu trainieren, so dass Neuronale Netze sich nicht für den Einsatz im FRF-Tool eignen.

Als Resultat der Validierung der Algorithmen und Testdaten ist festzuhalten, dass das Erkennen von Informationen mit Hilfe des J48- sowie des BayesNet-Algorithmus möglich ist.

Um einen Host zu klassifizieren werden alle Flow-Records eines Hosts selektiert und durch das FRF-Tool analysiert. Dem Host wird das Betriebssystem mit dem häufigsten Vorkommen zugewiesen. Unter Nutzung von Weka werden somit die Samples eines jeden Hosts einzeln aus der Datenbank geladen und gegen die Trainingsdatenbank untersucht.

Softwaredetektion

Die Übertragung des vorhergehend beschriebenen Verfahrens auf die Erkennung von Software zeigt verschiedene Probleme auf. Während bei der Detektion von Betriebssystemen die Trainingsdaten mit relativ geringem Aufwand generiert werden können, ist es bei Software notwendig, die spezifische Kommunikation (zum Beispiel Updates) aus den Flow-Records herauszufiltern und mit den Informationen zu taggen.

Im Proof of Concept wird die Softwaredetektion für Anwendersoftware auf Basis der Flow-Records eines Windows 7 Hosts durchgeführt. An dieser Stelle wird erkennbar, dass die Abspaltung der Softwarekommunikation von der allgemeinen Kommunikation des Betriebssystems mit erheblichem Aufwand verbunden ist, da ohne Informationen über die Inhalte der Übertragung die Kommunikation der Software schwer zu erkennen ist.

6.6.2. Ergebnissicherung

Im Anschluss an die heuristische Untersuchung werden die über das System erhaltenen Informationen in der Datenbank dem Asset hinzugefügt, sofern hierbei keine Konflikte bestehen. Das Layout der im Proof of Concept verwendeten Asset-Datenbank ist in Abbildung 6.3 dargestellt. Hierbei werden – um eine Mehrfachverwendung zu ermöglichen sowie Redundanzen zu vermeiden – die Informationen normiert in der Datenbank vorgehalten. Dabei werden die erkennbaren Betriebssysteme, Dienste sowie Anwendungen jeweils in eigenen Tabellen gesichert. Das Ergebnis der Auswertung wird mit entsprechenden Verlinkungen zwischen den Assets und den Kategorien gesichert. Zudem wird beim Asset der Zeitpunkt, zu dem dieses zuletzt gesehen wurde, aktualisiert.

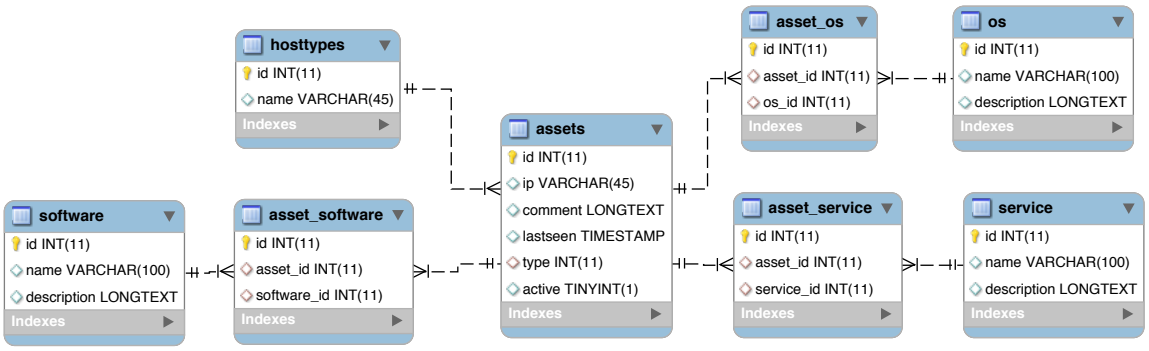


Abbildung 6.3.: Darstellung des Datenbanklayouts der Asset-Datenbank

6.7. Zusammenfassung

Basierend auf den vorhergehend beschriebenen Informationen lässt sich festhalten, dass die Klassifizierung der Hosts mit Hilfe von Bash-Skripten sowie heuristischen Verfahren möglich ist. Hierbei wird im Proof of Concept keine Computer-Nutzung durch den Anwender zugelassen.

Die Klassifizierung erlaubt mit einer relativ hohen Genauigkeit neben der Erkennung der Betriebssystemfamilie, also ob es sich um Windows, Linux oder Mac OS X handelt, auch die Major-Version des Betriebssystems zu detektieren. Die Erkennung der Systemart, also ob es sich um einen Server oder ein sonstiges Netz-Gerät handelt, liefert auf der Bedingung der Anzahl der eingehenden Verbindungen eine im Testnetz sehr hohe Genauigkeit. Eine Übertragung der Host-Klassifizierung auf ein aktiv genutztes Netz ist unter Umständen jedoch nicht ohne Anpassungen möglich. So müssen gegebenenfalls noch weitere Kriterien für die Erkennung von Servern sowie umfangreichere Trainingsdaten genutzt werden.

Die Dienst-Klassifizierung unter Nutzung der Standardports zeigte im Test eine perfekte Erkennungsrate, solange die Dienste auch auf den Standardports liefen. Ein Webserver, der beispielsweise auf Port 8080 läuft, konnte nicht klassifiziert werden. Hier ist eine Optimierung nicht ohne das Wissen über den Inhalt (zum Beispiel das genutzte Protokoll auf ISO/OSI-Schicht 5 oder höher) von Paketen möglich.

Bei der Softwareklassifizierung wird jedoch deutlich, dass sich eine Klassifizierung auf Basis von Flow-Records allein schwierig gestaltet. Zum Einen ist die Sammlung und Generierung geeigneter Trainingsdaten äußerst aufwändig (Generierung, Trennen von Anwendungs- und Systemdaten, etc.) zum Anderen stellt die Vielzahl an möglicher Software eine weitere Komplikation dar. Sofern Anwendungen mit festen Server-IP-Adressen, wie beispielsweise Dropbox arbeiten, lässt sich eine statische Erkennung durchführen.

Zusammengefasst ist festzuhalten, dass mit Hilfe von heuristischen Verfahren Betriebssysteme mit einer hohen Genauigkeit erkannt werden können und dass die Analyse der eingehenden IP-Adressen Aufschluss über die betriebenen Dienste geben kann. Die Erkennung der eingesetzten Software ist jedoch nicht mit einer ausreichenden Genauigkeit möglich gewesen.

7. Evaluation

Im Rahmen dieses Kapitels werden die Ergebnisse der Evaluation des FRF-Tools vorgestellt und mit den Gesamtanforderungen aus Kapitel 3 abgeglichen. Hierfür wird zunächst die Aufgabenstellung kurz zusammengefasst.

Das Ziel des im Rahmen dieser Arbeit konzeptionierten und prototypisch implementierten FRF-Tools ist es, auf Basis von Flow-Records Informationen über Hosts, Betriebssystem, angebotene Dienste sowie genutzte Software zu liefern. Hierfür werden in Kapitel 5 das Konzept des FRF-Tools beschrieben sowie nutzbare Verfahren skizziert. Unter Einhaltung der Gesamtanforderungen wurden sowohl ein Prototyp des FRF-Tools implementiert als auch ein geeignetes Labornetz zur Gewinnung von Trainings- und Evaluationsdaten aufgebaut und umgesetzt.

Für die Evaluation des FRF-Tools ist es notwendig, eine Datenbank mit Flow-Records sowie den zugehörigen Informationen über die betroffenen Hosts zu erstellen. Die Daten für diese Datenbank wurden mit Hilfe des Labornetzes gesammelt. Hierfür wurde die Hintergrundkommunikation der installierten Betriebssysteme, also insbesondere das automatische Abrufen und Installieren von Updates sowie Netzkommunikation von Systemdiensten, aufgezeichnet. Die Aufzeichnung der Flow-Records erfolgte im Edge-Router des Testnetzes, wobei alle Hosts am selben Switch wie der Router angeschlossen sind. Diese direkte Verbindung ermöglicht es, die Daten ohne Beeinflussung (zum Beispiel durch eine geänderte Maximum Transmission Unit (MTU)) durch andere Geräte zu erfassen.

Im Labornetz wurden Hosts mit den Betriebssystemen Windows 7, Windows 10, Debian 7, Debian 8, Ubuntu 14.04 Desktop, Ubuntu 14.04 Server, Ubuntu 16.04 Server, OpenSuse 12.2 sowie Mac OS X 10.12 (Sierra) eingesetzt. Für die vorhergehend genannten Betriebssysteme wurden über einen Zeitraum von 10 Tagen Flow-Records gesammelt und mit dem eingesetzten Betriebssystem getaggt.

Für die Erkennung der installierten Software wurde das Labornetz in seiner Größe auf lediglich zwei Clients reduziert. Die Client-Anwendungen Dropbox, OpenOffice, Skype und TeamViewer wurden auf einem Host mit Windows 7 als Betriebssystem, die Server-Anwendungen Open-SSH und MySQL auf einem Host mit Debian 8 als Betriebssystem installiert. Die Flow-Records der Anwendersoftware wurden für spezifische Aktionen wie zum Beispiel dem Abruf von Updates durch die Anwendung selbst oder Nutzung der Software (beispielsweise Synchronisation der Dropbox) aufgezeichnet. Die Records der Server-Anwendungen wurden durch eingehende Verbindungen mit unterschiedlicher Länge sowie verschiedenen Inhalten aus dem Internet generiert.

7.1. Beschreibung der Evaluation

Die im Testnetz, wie im vorhergehenden Abschnitt beschrieben, gewonnenen Daten werden für die Evaluation des FRF-Tools genutzt. Die Evaluation selbst geschieht in zwei Schritten. Im ersten Schritt wird die Klassifikation auf Basis der fest codierten Regeln für Hostart und Dienst durchgeführt. Die Ergebnisse dieser Evaluation werden mit den erwarteten Ergebnissen verglichen. Im zweiten Schritt wird die Klassifikation von Betriebssystem und Software basierend auf heuristischen Verfahren geprüft. Hierbei basiert die Validierung der heuristischen Verfahren auf einer Kreuz-Validierung mit zehn Faltungen. Dabei wird jeweils ein Teil der Daten für das Training des Algorithmus genutzt, der andere (kleinere) Teil für die Klassifizierung verwendet. Das Ergebnis der Klassifizierung wird im Anschluss mit der tatsächlichen Klasse verglichen und der prozentuale Fehler bestimmt. Die Daten werden so lange rotiert, bis alle Daten einmal für die Klassifizierung genutzt wurden. Hierbei ist es wichtig festzuhalten, dass nicht erlernte Daten auch nicht erkannt werden können und dadurch immer der am besten passenden trainierten Klasse zugewiesen werden. Die Durchführung der heuristischen Evaluation wird mit Hilfe des Dataminingtools Weka durchgeführt.

7.2. Ergebnisse der Evaluation

Nachfolgend werden die Ergebnisse der Evaluation zusammengetragen und hierbei nach Host- (und Betriebssystem-), Dienst- und Softwareklassifizierung dargestellt.

7.2.1. Hostklassifizierung

Aus der ersten Evaluation der regelbasierten Hostklassifizierung geht hervor, dass der im Testnetz als Webserver betriebene Server durch die Regeln der Host-Klassifizierung korrekt als Server klassifiziert werden konnte sowie kein Rechner fälschlicherweise als Server erkannt wurde. Somit ist die regelbasierte Hostklassifizierung auf Basis der eingehenden Verbindungen für die Erkennung von Servern geeignet.

Die weiterführende Evaluation der heuristischen Betriebssystemklassifizierung zeigt ein durchweg positives Ergebnis. Die Resultate der Klassifizierung mit verschiedenen Verfahren aus den Kategorien Bayes sowie Entscheidungsbäume sind in Anhang C beigelegt. Aus der Durchführung der Kreuzvalidierung geht hervor, dass sich der Algorithmus des BayesNet am besten eignete, um die trainierten Klassen wiederzuerkennen. Die BayesNet Klassifizierung erlaubte auf den Tests eine korrekte Klassifizierung von über 95% der Flow-Records, so dass eine sehr sichere Zuordnung des Betriebssystems zu einzelnen Hosts möglich ist. Betrachtet man die in Listing 7.1 aufgezeigten Ergebnisse, so wird erkennbar, dass auch eine Unterscheidung der einzelnen Betriebssystemversionen mit einer hohen Wahrscheinlichkeit möglich ist.

Neben dem BayesNet haben sich Entscheidungsbäume (vgl. Listing 7.2) ebenfalls als geeignetes Verfahren zur Erkennung des Betriebssystems bestätigt. Der J48 Entscheidungsbaum des Data Mining Tools Weka, der auf dem ID3 Entscheidungsbaum basiert, liefert mit über 94% korrekter Klassifizierungen ebenfalls sehr gute Ergebnisse, wobei der „relative absolute error“ deutlich höher liegt. Insbesondere bei Betriebssystemen, zu denen eine geringe Anzahl an Trainingsdaten vorhanden war (zum Beispiel debian 7 und debian8), war ein erhöhter Anteil an Fehlklassifizierungen feststellbar (vgl: Listing 7.2).

Listing 7.1: Ergebnis der Kreuzvalidierung mit BayesNet

```
=== Summary ===
Correctly Classified Instances       14451               95.7273 %
Incorrectly Classified Instances       645                4.2727 %
Kappa statistic                          0.9397
Mean absolute error                      0.0103
Root mean squared error                  0.0872
Relative absolute error                  6.5372 %
Root relative squared error             31.1564 %
Total Number of Instances            15096

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
 1513    0   80   41   12   97    8   15    0 |    a = Mac OSX Sierra
    0  717    1    7   10    3   33   14    0 |    b = ubuntu16Server
    3    3  189    4   22    4    4    3    0 |    c = ubuntu14Desktop
   30    7    1 6088    0  131    1    0    0 |    d = Windows10
    0    0    4    0   23    1    1    1    0 |    e = ubuntu14Server
   32    0    0    1    0 1179    1    0    1 |    f = Windows7
    1    0   18   17    5    3 4715    1    0 |    g = Open Suse 12
    0    0    1    0    0    3    4   25    2 |    h = debian8
    0    0    0    5    1    0    0    8    2 |    i = debian7
```

Listing 7.2: Ergebnis der Kreuzvalidierung mit J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       14233               94.2833 %
Incorrectly Classified Instances       863                5.7167 %
Kappa statistic                          0.9177
Mean absolute error                      0.0467
Root mean squared error                  0.1214
Relative absolute error                 29.796  %
Root relative squared error             43.3675 %
Total Number of Instances            15096

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
 1443    0    1  154    0  168    0    0    0 |    a = Mac OSX Sierra
    0  727    4   18    3    0   33    0    0 |    b = ubuntu16Server
    3    9  149   11    2    0   58    0    0 |    c = ubuntu14Desktop
    4    0    0 6220    0   32    2    0    0 |    d = Windows10
    0    5    5    0    5    1   14    0    0 |    e = ubuntu14Server
   15    0    0  263    0  933    3    0    0 |    f = Windows7
    0    0    2   11    0    8 4739    0    0 |    g = Open Suse 12
    0    0    0    2    0    0   17   13    3 |    h = debian8
    0    0    0    2    0    0    6    4    4 |    i = debian7
```

7.2.2. Dienstklassifizierung

Die Erkennung der betriebenen Dienste zeigte im Testnetz sowohl positive als auch negative Resultate. Die auf den Standardports betriebenen Dienste konnten fehlerfrei erkannt werden. Dienste wie ein auf Port 8080 betriebener Webserver, ein auf Port 8085 betriebenes Wiki oder ein auf Port 60022 betriebener SSH-Server ließen sich durch die statischen Regeln nicht erkennen (vgl. Tabelle 7.1).

Tabelle 7.1.: Ergebnis der Dienstklassifizierung

Dienst	Genutzter Port	Erkennungsresultat
FTP-Server	20/21	erkannt
FTP-Sever	60021	nicht erkannt
SSH	22	erkannt
SSH	60022	nicht erkannt
Mailserver	25/110/143	erkannt
Mailserver (Imap)	443	falsch als Webserver erkannt
DNS-Server	53	erkannt
Webserver	80	erkannt
Webserver	443	erkannt
Webserver	8080	nicht erkannt
Webserver (javabasiertes Wiki)	8085	nicht erkannt

Zusammengefasst ist feststellbar, dass eine Dienstklassifizierung, die rein auf einer statischen Port-Dienst-Zuordnung basiert, nur unzureichende Ergebnisse liefert. Aus diesem Grund ist es hier notwendig, weitere Nachbesserungen am Verfahren durchzuführen beziehungsweise die Erweiterung um zusätzliche Algorithmen vorzunehmen.

7.2.3. Softwareklassifizierung

Basierend auf den Ergebnissen der Dienstklassifizierung ist es von Interesse eine systematische und heuristische Analyse der Flow-Records hinsichtlich der eingesetzten Software vorzunehmen. Die heuristische Untersuchung eignet sich hier, um analog zur Betriebssystemdetektion die eingesetzte Software auch bei Unsicherheit erkennen zu können.

Bei der Auswertung der verschiedenen Klassifikatoren ist erkennbar, dass die Ergebnisse deutlich differieren. So zeigt sich bei den meisten Verfahren eine Erkennungsrate von ca. 50%. Das beste Ergebnis für die Softwaredetektion liefert der BFTree mit einer Erkennungsrate von über 88%. Der BayesNet Klassifier lässt sich nicht für die Softwareklassifizierung einsetzen, da beim Versuch der Klassifizierung die Standardverteilung einzelner Programme als zu niedrig bewertet wird (vgl. Listing 7.3).

Listing 7.3: Ergebnis der Kreuzvalidierung mit BFTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         467               88.9524 %
Incorrectly Classified Instances        58               11.0476 %
Kappa statistic                          0.7859
Mean absolute error                      0.0454
Root mean squared error                  0.1782
Relative absolute error                 25.2083 %
Root relative squared error             59.6132 %
Total Number of Instances              525

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
 118   0   4  12   0   0 |   a = Dropbox
   0   7   0   0   0   1 |   b = SSH
   9   0   2  12   0   0 |   c = TeamViewer
   7   0   4 319   0   0 |   d = Skype
   7   0   0   2   5   0 |   e = OpenOffice
   0   0   0   0   0  16 |   f = MySQL
```

Nach der Kreuzvalidierung der reinen Software-Flow-Records, also ohne die Flow-Records der Betriebssysteme, ist es notwendig, diese Untersuchung ebenfalls auf dem vollständigen Set an Flow-Records durchzuführen. Hierbei sind sowohl die Update-Anfragen und Kommunikation der Betriebssysteme als auch die Kommunikation der zu klassifizierenden Software vorhanden. Hier wird erkennbar, dass Software zwar zum Teil erkannt wird, aber bis auf Skype und Dropbox meist eine Fehlklassifikation stattfindet (vgl. Listing 7.4).

Listing 7.4: Ergebnis der Kreuzvalidierung mit BayesNet

```
=== Summary ===

Correctly Classified Instances        2663               92.5939 %
Incorrectly Classified Instances       213                7.4061 %
Kappa statistic                          0.9031
Mean absolute error                      0.0115
Root mean squared error                  0.0861
Relative absolute error                 11.2946 %
Root relative squared error             38.2524 %
Total Number of Instances             2876

=== Confusion Matrix ===
a = Dropbox                                     i = ubuntu16Server
b = SSH                                         j = Open Suse 12
c = TeamViewer                                  k = debian8
d = Skype                                       l = debian7
e = OpenOffice                                  m = Windows10
f = MySQL                                       n = ubuntu14Server
g = ubuntu14Desktop                             o = Windows7
h = Mac OSX Sierra
```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	<-- classified as
39	0	0	3	1	0	0	0	0	0	0	0	4	0	0	a
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	b
1	0	0	1	0	0	0	0	0	0	0	0	5	1	0	c
3	0	1	94	0	0	0	1	2	0	0	0	5	0	10	d
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	e
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	f
0	0	0	0	0	0	41	3	2	0	0	0	2	0	0	g
0	0	0	0	0	0	1	186	5	0	0	0	4	1	6	h
0	0	0	0	0	0	10	1	246	2	1	0	5	0	1	i
0	0	0	0	0	0	2	3	0	989	0	0	5	0	0	j
0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	k
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	l
0	0	0	0	0	0	0	13	0	0	0	0	257	0	9	m
0	0	0	0	0	0	6	0	0	0	0	0	0	5	0	n
13	0	0	8	0	0	0	8	0	1	0	0	55	0	800	o

7.2.4. Zusammenfassung

Zusammengefasst ist festzuhalten, dass sich mit Hilfe von Flow-Record-Fingerprinting unter Idealzuständen aussagekräftige Informationen über die im Netz befindliche Hosts sammeln lassen. Ebenfalls gilt dies für Betriebssysteme. Die Erkennung von Diensten ist abhängig von verschiedenen Faktoren, während die Klassifizierung von eingesetzter Software nahezu unmöglich ist.

7.3. Abgleich der Resultate mit den Anforderungen

In diesem Abschnitt werden die in Kapitel 3 zusammengestellten Gesamtanforderungen mit den Ergebnissen des evaluierten FRF-Tools abgeglichen.

7.3.1. Funktionale Anforderungen

FA 1 Heterogenität
Das im Rahmen dieser Arbeit entwickelte FRF-Tool ist in heterogenen Netzen, also mit unterschiedlicher Hard- und Software einsetzbar. Diese Kompatibilität wird durch die im Labornetz einsetzte Software sowie die genutzte Hardwarevirtualisierung in Verbindung mit den vorhergehend beschriebenen Evaluationsergebnissen belegt.

FA 2 Weitläufigkeit
Die im Konzept des FRF-Tools beschriebene Erfassung der Flow-Records in Knotenpunkten des Netzes sowie das Zusammenführen der gewonnenen Daten mit Hilfe verschiedener Flow-Sammler in der Nähe der verschiedenen Flow-Recorder erlauben den Einsatz in weitläufigen Netzen. Die Zusammenführung der Daten in einer zentralen Datenbank erlaubt es auch bei (geographisch) weit verteilten Flow-Recordern vollständige Daten zu erfassen. Aus diesem Grund ist die Anforderung der Einsetzbarkeit in weitläufigen Netzen erfüllt.

FA 3 Grenzerfassung
Die Erfassung von Flow-Records in Grenzpunkten zu autonomen Netzen ist analog zu der Erfassung der Flow-Records im Edgerouter des Labornetzes möglich. Findet im Grenzpunkt jedoch eine Network Address Translation (NAT) statt, so ist es nicht mehr möglich, die Flow-Records zu einzelnen Hosts zuzuordnen. Da dennoch Aussagen über das autonome Netz als Ganzes getroffen und die Datenerfassung im Grenzpunkt durchgeführt werden können, ist diese Anforderung als erfüllt anzusehen.

FA 4 Teilnehmerzahl und Traffic
Da moderne Enterprise-Netzhardware Flow-Records oftmals nativ erstellen kann, hierfür nur eine geringe Rechenleistung benötigt wird und die notwendige Datenmenge für die Übertragung der Records sehr gering ist, beeinflussen eine hohe Teilnehmerzahl oder viel Traffic die Nutzung des FRF-Tools nicht. Als Folge entstehen lediglich proportional zur Anzahl der Netzteilnehmer und des Trafficvolumens mehr Flow-Records. Somit ist das FRF-Tool auch in Netzen mit vielen Teilnehmern sowie hohem Traffic einsetzbar.

FA 5 Asseterkennung
Eine Erkennung verschiedener Betriebssysteme ohne Patchstand oder installierte Software wurde mit dem FRF-Tool erfolgreich getestet. Die in Anforderung FA 5 ebenfalls verlangte Erkennung von installierter Software und Versionsstand ist nicht vollständig gegeben, so dass diese Anforderung teilweise erfüllt ist.

FA 6 Netze Dritter
Da eine passive Analyse des durchgeleiteten Traffics, also basierend auf dem für den Flow-Recorder sichtbaren Traffic, durchführbar ist, wird die Anforderung der Kompatibilität mit Netzen Dritter erfüllt.

FA 7 Routing
So lange der zu analysierende Traffic den Flow-Recorder durchläuft, ist die Analyse möglich und die Anforderung somit erfüllt.

FA 8 Störungsfreiheit
Da lediglich eine passive Analyse des anfallenden Datenverkehrs erfolgt, ist die Anforderung der Störungsfreiheit erfüllt.

FA 9 Übertragungsinhalte
Theoretisch ist eine Erkennung von Übertragungsinhalten auf Basis der Metadaten der Flow-Records möglich, ließ sich im Proof of Concept durch das FRF-Tool jedoch nicht zeigen, so dass diese Anforderung nicht erfüllt ist.

Tabelle 7.2.: Abgleich mit den Funktionalen Gesamtanforderungen

ID	Schlagwort	Anforderung	Priorität	Ergebnis
FA 1	Heterogenität	Das Tool muss in heterogenen Systemen einsetzbar sein.	++	erfüllt
FA 2	Weitläufigkeit	Das Tool muss über (geographisch) weitläufige Netze einsetzbar sein.	++	erfüllt
FA 3	Grenzerfassung	Die Datenerfassung muss in Grenzpunkten zu autonomen Netzen möglich sein.	++	erfüllt
FA 4	Teilnehmerzahl und Traffic	Das Tool muss in Netzen mit vielen Teilnehmern und hohem Traffic einsetzbar sein.	+++	erfüllt
FA 5	Asseterkennung	Das Tool muss eingesetzte Betriebssysteme sowie Software und deren Versionsstand erkennen.	+++	teilweise erfüllt
FA 6	Netze Dritter	Das Tool muss Assets auch in durch Dritte betreute Netze erkennen können.	++	erfüllt
FA 7	Routing	Die Analyse muss mit dynamischem sowie komplexem Routing kompatibel sein.	+	erfüllt
FA 8	Störungsfreiheit	Der Einsatz des Tools muss zuverlässig und zeitnah möglich sein, ohne dabei Störungen zu verursachen.	++	erfüllt
FA 9	Übertragungsinhalte	Das Tool muss Rückschlüsse auf Übertragungsinhalte ermöglichen.	+	nicht erfüllt

\+ Anforderung nice to have / ++ Anforderung wichtig / +++ Pflichtanforderung

7.3.2. Nicht Funktionale Anforderungen

NFA 1 Verfügbarkeit
Durch die Analyse erfolgt keine negative Beeinflussung der Verfügbarkeit betriebener Dienste und Anwendungen, so dass diese Anforderung erfüllt ist.

NFA 2 Performance
Da lediglich die Flow-Records zum Flow-Sammler hin übertragen werden müssen sowie die gesammelten Flow-Records vom Flow-Sammler in die Datenbank eingespielt werden, ist der anfallende Traffic so gering, dass keine negative Beeinflussung des Netzes sowie der Performance des Netzes erfolgt.

NFA 3 Auswertbarkeit
Die Resultate des FRF-Tools sowie das genutzte Datenbanklayout ermöglichen eine weitere Auswertung. Somit ist diese Anforderung ebenfalls erfüllt.

NFA 4 Kosten
Lizenzen oder ähnliche Kosten fallen für das FRF-Tool nicht an, da vollständig kostenlose und offene Software eingesetzt wird. Es werden lediglich Flow-Sammler, die Datenbank sowie der Analyseserver benötigt. Abhängig von der Anzahl der zu speichernden und zu verarbeitenden Daten kann es notwendig sein, leistungsstärkere Hardware zu beschaffen. Aus diesem Grund ist diese Anforderung teilweise erfüllt.

NFA 5 Vorhandene Infrastruktur
Für die Erfassung der Daten lassen sich die vorhandene Infrastruktur, also die bestehenden Gateways sowie die verbauten Switches nutzen, so dass diese Anforderung erfüllt ist.

NFA 6 IPv6
Eine Nutzung des vorgestellten Ansatzes mit IPv6 wurde im Proof of Concept nicht getestet, ist jedoch bei Nutzung von aktueller Hard- und Software zur Erstellung der Flow-Records ohne Weiteres möglich, so dass diese Anforderung als erfüllt anzusehen ist.

NFA 7 Anonymisierung
Da im Konzept des FRF-Tools eine Anonymisierung vorgesehen ist und mit Hilfe der Datenbank keine schützenswerten Informationen der Clients an das FRF-Tool übermittelt werden, ist die Anonymisierung der Daten als erfüllt anzusehen. Da jedoch keine Anonymisierung der Flow-Records vor dem Import in die Datenbank möglich ist, gilt die Anforderung als nur teilweise erfüllt.

NFA 8 Datenschutz
Da schützenswerte Daten automatisch entfernt werden und Informationen für Unbefugte ohne Mehraufwand krimineller Natur nicht ersichtlich sind, gilt die Anforderung der Einhaltung des Datenschutzes als erfüllt.

NFA 9 Gesetzeskonformität
Da keine Manipulation sowie keine Analyse der Übertragungsinhalte erfolgt, ist die im FRF-Tool implementierte Lösung als gesetzeskonform im Sinne der NFA 9 anzusehen.

NFA 10 Transparenz
Da der vorgestellte Ansatz ein passives Verfahren ist, welches den Netzverkehr nicht verändert, wird die Anforderung der Transparenz erfüllt.

NFA 11 Selektion
Eine Selektion einzelner Hosts ist bei der Erfassung von Flow-Records nicht möglich, so dass initiale Daten aller Hosts erfasst werden. Die Selektion ist im Nachhinein mit Hilfe der Datenbank möglich. Somit ist diese Anforderung als teilweise erfüllt anzusehen.

Tabelle 7.3.: Abgleich mit den Nicht Funktionalen Gesamtanforderungen

ID	**Schlagwort**	**Anforderung**	**Priorität**	**Ergebnis**
NFA 1	Verfügbarkeit	Die Erreichbarkeit, Performance sowie Wartbarkeit bereitgestellter Dienste und Server darf nicht eingeschränkt werden.	+++	erfüllt
NFA 2	Performance	Die Performance und Leistungsfähigkeit des Netzes darf nicht eingeschränkt werden.	+++	erfüllt
NFA 3	Auswertbarkeit	Gewonnene Daten müssen für andere Systeme lesbar und auswertbar sein.	+	erfüllt
NFA 4	Kosten	Zusätzliche Kosten durch Lizenzen oder neue Hardware sind zu vermeiden.	+	teilweise erfüllt
NFA 5	Vorhandene Infrastruktur	Vorhandene Infrastruktur und gegebene Möglichkeiten sind zu bevorzugen.	+	erfüllt
NFA 6	IPv6	Das Tool muss moderne Protokolle (IPv6) unterstützen.	++	erfüllt
NFA 7	Anonymisierung	Erfasste Daten müssen geeignet anonymisiert sein, um den Datenschutz zu gewährleisten.	+++	teilweise erfüllt
NFA 8	Datenschutz	Bei der Datenerfassung muss der Datenschutz im Sinne des Bundesdatenschutzgesetzes (BDSG) sowie des Bayerisches Datenschutzgesetzes (BayDSG) eingehalten werden.	+++	erfüllt
NFA 9	Gesetzeskonformität	Die Datenerfassung sowie die Analyse müssen die Regelungen des StGB einhalten.	+++	erfüllt
NFA 10	Transparenz	Das Tool muss für Netzteilnehmer transparent sein.	+++	erfüllt
NFA 11	Selektion	Eine gezielte Auswahl der zu überwachenden und analysierenden Systeme muss vorab möglich sein.	+	teilweise erfüllt

+ Anforderung nice to have/ ++ Anforderung wichtig / +++ Pflichtanforderung

7.3.3. Zusammenfassung

Der Abgleich des FRF-Tools mit den im Rahmen dieser Arbeit entwickelten Anforderungen zeigt eine fast vollständige Erfüllung dieser Gesamtanforderungen. In Abbildung 7.1 wird dies graphisch veranschaulicht. Im Vergleich zu den in Kapitel 4 vorgestellten themenverwandten Arbeiten wird deutlich, dass das FRF-Tool ein deutlich besseres Resultat als die vorgestellten Verfahren liefert.

Nicht erfüllt wird lediglich die Anforderung der Erkennung von Übertragungsinhalten. Teilweise erfüllt werden die Anforderung der gezielten Selektion, Anonymisierung der Daten, Vermeidung zusätzlicher Kosten sowie die vollständige Asseterkennung.

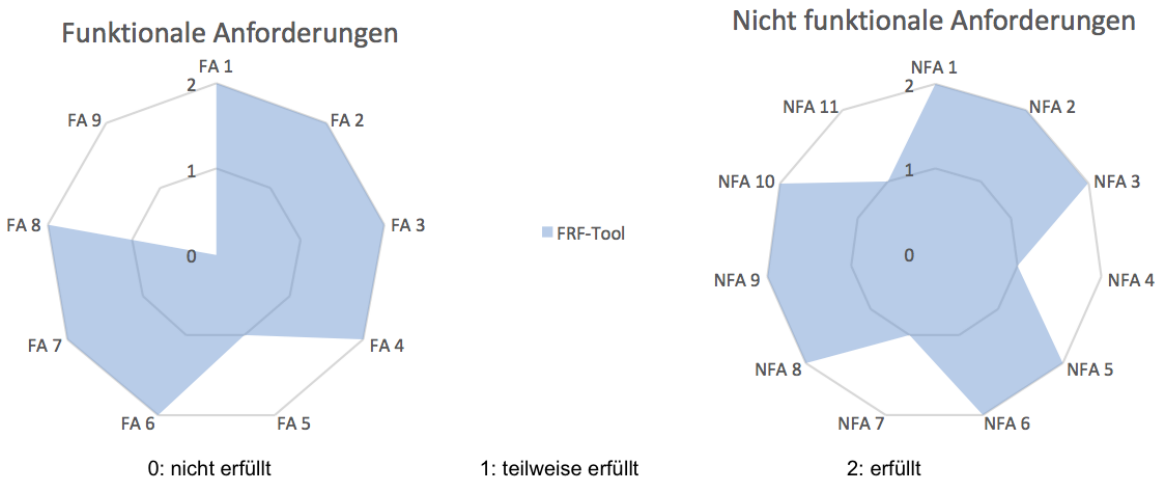


Abbildung 7.1.: Visualisierung der Anforderungserfüllung

8. Resümee und Ausblick

In diesem Kapitel wird ein Resümee über das im Rahmen dieser Arbeit untersuchte Konzept des FRF-Tools gezogen. Anschließend folgt ein Ausblick über die möglichen Verbesserungen am vorgestellten Konzept.

8.1. Resümee

Der in dieser Arbeit gewählte Ansatz geht als Datenbasis von den leichtgewichtigen Flow-Records aus und entwickelt mittels dieser ein geeignetes Konzept, welches in einem ersten Praxisschritt bereits in einem Testsystem konfiguriert und implementiert wurde. Durch Nutzung dieses Testsystems liegen erste aussagefähige Informationen vor, die wiederum mit den Gesamtanforderungen verglichen werden konnten.

Mit Hilfe von Kapitel 1 wurde eine allgemeine Einleitung in das Thema dieser Arbeit gegeben. Hierbei wurden ebenfalls kurz die Nachteile der bisherigen Verfahren angerissen. Im Anschluss wurden die zu erfüllenden Teilziele präsentiert. In Kapitel zwei folgte eine Vorstellung verschiedener wesentlicher Grundlagen. Die Funktionalen und Nicht Funktionalen Anforderungen an das im Rahmen dieser Arbeit entwickelte und mit Hilfe des Prototyps evaluierten Konzepts wurden in Kapitel 3 erarbeitet. Hierfür wurden die drei Szenarien des Hochschulrechenzentrums, der Unternehmen sowie der Strafverfolgungsbehörden und Nachrichtendienste herangezogen, um individuelle Anforderungen abzuleiten. Die individuellen Anforderungen der drei Szenarien wurden im Anschluss in die Gesamtanforderungen dieser Arbeit überführt und priorisiert. In Kapitel 4 wurden verschiedene aktive, passive sowie ein hybrides Verfahren vorgestellt und mit den Gesamtanforderungen aus Kapitel 3 abgeglichen. Hierbei wurde ersichtlich, dass keines der bisherigen Verfahren in der Lage war, die Anforderungen vollumfänglich zu erfüllen. In Kapitel 5 wurde das Konzept des FRF-Tools entwickelt und präsentiert. Im ersten Abschnitt wurden die Datenerfassung, -Übertragung und -Speicherung vorgestellt und hierbei auch auf die Anforderungen des Datenschutzes eingegangen. Der zweite Abschnitt beschreibt die Klassifizierung der gewonnenen Daten und lieferte hierfür auch die notwendigen Definitionen. Der letzte Abschnitt erläuterte die Auswertung der Daten. Hierfür wurden die verfügbaren Datenfelder sowie ein Verfahren zur Auswahl der am besten geeigneten Datenfelder und Daten präsentiert. Ebenfalls wurden in diesem Abschnitt die heuristischen Verfahren der Bayes-Klassifikation, der Entscheidungsbäume sowie der neuronalen Netze allgemein beschrieben. Abschließend wurde die Auswahl eines geeignetes heuristischen Verfahrens erläutert. Im Anschluss erfolgte in Kapitel 6 die Beschreibung der Konfiguration und Implementierung. Hierbei wurde sowohl auf die Grundlagen des Proof of Concept als auch auf das eingesetzte Labornetz sowie die Gewinnung der Trainingsdaten, die Erfassung, Übertragung und Speicherung der Daten und die Datenklassifizierung eingegangen. Abschließend wurde die Datenauswertung mit der Datensicherung erläutert. In Kapitel 7 wurden die Ergebnisse der Evaluation beschrieben und hierbei festgestellt, dass die Betriebssystemklassifikation gute Ergebnisse, die Software- und Dienstklassifikation jedoch verbesserungswürdige Resulate lieferten. Abschließend wurden die Funktionalen und Nicht

Funktionalen Gesamtanforderungen mit dem Konzept abgeglichen und hierbei festgestellt, dass ein Großteil der Gesamtanforderungen – jedoch nicht alle – erfüllt werden konnten.

Die genauere Betrachtung der Resultate aus Kapitel 7 macht deutlich, dass weitere Anpassungen notwendig sind, um die Gesamtanforderungen dieser Arbeit besser erfüllen zu können. Insbesondere sind diese Verbesserungen hinsichtlich der Erkennung des Updatestandes sowie bei der Wiedererkennung installierter Software nötig. Das genutzte Verfahren zur Diensterkennung zeigt ebenfalls Potential für Anpassungen, durch die die Ergebnisqualität verbessert werden kann. Trotz der vorhergehend genannten Einschränkungen bezüglich Updatestand, installierter Software und Dienste ist es möglich, das vorgestellte Konzept in einem Feldversuch auch außerhalb der Virtualisierung des Labornetzes einzusetzen und dabei zufriedenstellende Ergebnisse zu erhalten.

8.2. Ausblick

Nachfolgend wird eine Auswahl über im Nachgang notwendige Erweiterungen gegeben, die zielführend die Entwicklung und Forschung mittels Flow-Records voranbringen können.

Das Testsystem des Proof of Concept bestand lediglich aus vollvirtualisierten Rechnern, die über das gleiche Medium am Gateway angebunden waren. Da es keine Möglichkeit gab physikalische Geräte mit dem Netz zu verbinden, gilt es diese im nächsten Schritt einzubringen. Zudem ist die Netzinfrastruktur (Switches, Gateways, etc.) hinsichtlich Art, Hersteller, Alter, Anzahl sowie Kapazität zu variieren.

In der Testumgebung waren alle Rechner über einen Switch direkt am Gateway, das als Flow-Generator diente, angeschlossen. Es erfolgte keine Evaluation der Auswirkung von unterschiedlichen Übertragungsmedien mit unterschiedlichen Bandbreiten. Hierbei können verschiedene Übertragungsarten wie W-LAN, Netzwerkkabel sowie DSL-Verbindungen genutzt werden.

Insbesondere erlaubt die Nutzung von W-LAN die Anbindung von Mobilfunkgeräten, Tablets oder weiteren Systemen, welche nicht über kabelgebundene Netzwerkanschlüsse verfügen. Neben einer Erweiterung um mobile Endgeräte ist es auch nötig, verschiedene Rechnerarten (Hersteller, Alter, etc.) einzusetzen. Hierdurch wird es möglich, weitere qualitativ hochwertige Daten zu generieren. Insbesondere geht es beim Hinzufügen weiterer Systeme um die Erweiterung der erkennbaren Betriebssysteme, Dienste und Anwendungen.

Wie erwähnt befindet sich das Gateway in unmittelbarer Nähe zu allen angeschlossenen Systemen. Hier ist es daher notwendig, die Auswirkungen der Distanz zwischen Flow-Recorder und Zielsystem auf die Erkennungsrate zu untersuchen, um die Gesamtanforderung der Kompatibilität mit (geographisch) weitläufigen Netzen zu untermauern.

Neben der Auswirkung der Distanz sollte zudem untersucht werden, wie sich der Einsatz von NAT, also mehrere verschiedene Rechner, die unter der gleichen IP-Adresse erfasst werden, auf die Qualität der Auswertung auswirkt.

Wichtig ist darauf hinzuweisen, dass die Datensammlung im Rahmen dieser Arbeit rein auf der Kommunikation des Betriebssystems sowie der Anwendungen basiert. Hierbei fand explizit keine normale Nutzung des Anwenders wie beispielsweise Internet surfen statt. Aus diesem Grund ist es notwendig zu untersuchen, in wie weit sich diese zusätzlich entstehenden Flow-Records auf die Qualität der Untersuchungsresultate auswirken.

Im Rahmen der Datenauswertung wurde eine gezielte Auswahl an Flow-Records getroffen. Generell sollte die Auswahl der für das Training sowie die Analyse genutzten Flow-Records

weiter untersucht und optimiert werden. Hierbei kann es sich sowohl um eine Einschränkung der Auswahl als auch um eine Erhöhung der Anzahl der Flow-Records handeln.

Alle Trainingsdaten wurden im Proof of Concept selbst generiert. Da dies sehr aufwändig und auf Grund manueller Tätigkeit fehleranfällig ist, bietet es sich an, den Aspekt der Nutzung von externen Datenquellen zu untersuchen. Hierzu zählen beispielsweise Datenbanken mit IP-Eigentümer-Zuordnung.

Zudem scheint es notwendig zu sein, für die unterschiedlichen Auswertungen von Betriebssystem, Diensten und Software jeweils eigene Sets an Flow-Records einzusetzen. Dies bedingt eine Prüfung der jeweiligen Auswahl. Ein Beispiel der Hauptprobleme des FRF-Tools, welches sich auch in der schlechten Erkennungsrate von Software widerspiegelt, stellt das Hintergrundrauschen des Betriebssystems dar. Dieses Hintergrundrauschen muss beim Tagging von den Daten der Anwendung selbst getrennt werden.

Als weiteres Problem im Rahmen der Flow-Record-Analyse hat sich die zunehmende Verbreitung von Content Delivery Networks herausgestellt, welche oft der Verteilung von Dateien dienen. Hauptproblem der Nutzung der CDNs ist, dass basierend auf der IP-Adresse keine Aussagen über den potentiellen Inhalt einer Übertragung (z.B. Updates) getroffen werden können. Um dieser Herausforderung zu begegnen, bietet sich die Nutzung von ausgewählten Paketinhalten an. Derartige Paketinhalte können sowohl die ersten 50 Byte eines Pakets als auch die Inhalte von DNS-Anfragen sein.

Auch hat sich im Rahmen der Untersuchung gezeigt, dass bei verschiedenen Betriebssystemen, Diensten oder Anwendungen Unsicherheiten bei der Klassifizierung bestehen können. Um diesen Unsicherheiten entgegenzuwirken bietet es sich an, die Spannbreite der verwendeten Trainingsdaten zu erhöhen. Diese Erhöhung geht mit der Nutzung zusätzlicher Betriebssysteme sowie Anwendungen einher.

Eine weitere Möglichkeit den Unsicherheiten der Klassifizierung zu begegnen, stellt die Option dar, aktiv Anfragen an das zu klassifizierende System zu stellen. Beispiele für derartige Anfragen sind der Aufbau von Verbindungen oder das Senden von Paketen und Auswerten der Antworten des Zielsystems.

Da im Rahmen dieser Arbeit lediglich ein kleiner Teilausschnitt der heuristischen Verfahren hinsichtlich deren Eignung für die Flow-Record-Analyse untersucht werden konnte, ist es notwendig, weitere Verfahren zu evaluieren. Zusätzlich zur Evaluation weiterer Verfahren gilt es, die Kombination verschiedener Verfahren zu prüfen. Hier sei auch noch einmal darauf hingewiesen, dass unterschiedliche heuristische Verfahren für unterschiedliche Erkennungsszenarien (Betriebssystem, Dienste, Software) eingesetzt werden können und somit mit einer unterschiedlichen Auswahl an Flow-Records kombiniert werden können.

Insbesondere ist zu untersuchen, welche Anpassungen an der genutzten Datenbasis notwendig sind, um neuronale Netze für den Prozess der heuristischen Analyse einsetzen zu können. Dies ist sinnvoll, da neuronale Netze den Vorteil bieten, auch bei ungenauen Daten eine oftmals gute Klassifizierung treffen zu können.

Des Weiteren bietet es sich an, die heuristischen Verfahren mit Hilfe von regelbasierten Entscheidungen zu kombinieren.

Die einzelnen Schritte erfolgten im Proof of Concept in Einzelanwendungen oder mit Hilfe von Bash-Skripten. Für eine produktive Nutzung ist es unbedingt notwendig, diese Einzelschritte in eine zusammenhängende Anwendung zusammenzuführen.

Generell ist festzuhalten, dass in dieser Arbeit für den Proof of Concept ein hoher manueller Aufwand notwendig war. Für den Einsatz in größeren Umgebungen ist es hier zwingend notwendig, eine automatisierte Lösung zu fokussieren. Dies kann auch in Teilschritten erfolgen.

A. Ansible

A.1. Flow-Recorder und Router

Das nachfolgende Ansible-Skript führt die Installation und Konfiguration des Routers mit mehreren Netzinterfaces aus. Auf eth1 ist hierbei das zu überwachende Netz hinterlegt. Angeschlossene Geräte erhalten dynamische IP-Adressen aus dem Netz 10.150.254.0/24. Als Betriebssystem wird Debian Jessie eingesetzt.

Listing A.1: Konfiguration des Routers mit Ansible

```
1  ---
2  - hosts: flowrouter
3    sudo: yes
4    tasks:
5      - name: Install Softflowd
6        apt: name=softflowd state=installed update_cache=true
7      - name: Install Python Mysql
8        apt: name=python-mysqldb state=installed update_cache=true
9      - name: Mysql-Server
10       apt: name=mysql-server state=installed update_cache=true
11     - name: create DB
12       mysql_db: name=flows  state=present
13     - name: copy import data
14       copy: src=dump.sql dest=/tmp
15     - name: import DB
16       mysql_db: name=flows state=import target=/tmp/dump.sql
17     - name: create DB User
18       mysql_user: name=thesis password=thesis priv=*.*:ALL,GRANT state=
             present
19     - name: Install flow-tools
20       apt: name=flow-tools state=installed update_cache=true
21     - name: Install screen
22       apt: name=screen state=installed update_cache=true
23     - name: Install iptables persistent
24       apt: name=iptables-persistent state=installed update_cache=true
25     - name: Install tree
26       apt: name=tree state=installed update_cache=true
27     - name: Install DHCP
28       apt: name=isc-dhcp-server state=installed update_cache=true
29       notify:
30         - restart dhcp
31     - name: DHCP Config
32       template: src=templates/dhcpd.conf.txt dest=/etc/dhcp/dhcpd.conf owner=
             root group=root mode=0644
33     - name: Creates data directory
34       file: path=/flowstore/data state=directory owner=root group=root mode
             =0775 recurse=yes
35     - name: Creates data directory
36       file: path=/flowstore/old state=directory owner=root group=root mode
             =0775 recurse=yes
```

```
37      - name: Record Startup Script
38        template: src=templates/startRecord.sh.txt dest=/root/startRecord.sh
              owner=root group=root mode=0744
39      - name: configure flow collector
40        template: src=templates/softflowd.txt dest=/etc/default/softflowd owner
              =root group=root mode=0744
41      - name: configure flow collector config
42        template: src=templates/flow-capture.conf.txt dest=/etc/flow-tools/flow
              -capture.conf owner=root group=root mode=0744
43        register: flow_config
44      - name: restart Flow collector deamon
45        service: name=flow-capture state=restarted
46        when: flow_config|changed
47      - name: Enable IPv4 forwarding
48        command: sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/g' /etc
              /sysctl.conf
49      - name: Enable IPv6 forwarding
50        command: sed -i 's/#net.ipv6.conf.all.forwarding=1/net.ipv6.conf.all.
              forwarding=1/g' /etc/sysctl.conf
51      - name: Forwarding Changes
52        command: sysctl -p /etc/sysctl.conf
53      - name: Enable IPv4 forwarding
54        command: iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
55      - name: Enable IPv4 forwarding
56        command: iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,
              ESTABLISHED -j ACCEPT
57      - name: Enable IPv4 forwarding
58        command: iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
59      - name: create import script
60        template: src=templates/doImport.sh.txt dest=/usr/local/bin/doImport.sh
               owner=root group=root mode=0774
61      - name: Install Importing Cronjob
62        cron: name="import␣data" minute="*/4" job="/usr/local/bin/doImport.sh␣>
              ␣/dev/null"
63    handlers:
64     - name: start softflowd
65       command: softflowd -i eth1 -n 127.0.0.1:4432
66
67     - name: restart dhcp
68       service: name=isc-dhcp-server state=restarted
```

Listing A.2: Import-Skript

```
#!/bin/bash
for i in $(find /flowstore/data/ -name "*ft-v*") ; do
        flow-export -f3 -u "thesis:thesis:localhost:3306:flows:flowinput" <
            $i ;
        mv $i /flowstore/old/$(basename $i);
done

mysql -uthesis -pthesis flows -e "DELETE␣FROM␣flowinput␣WHERE␣srcaddr␣not␣
    like␣'10.150.%'␣and␣dstaddr␣not␣like␣'10.150.%'"
mysql -uthesis -pthesis flows -e "DELETE␣FROM␣flowinput␣WHERE␣unix_secs␣<␣(
    UNIX_TIMESTAMP()-60*60*24*7)";


mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dns-response'␣WHERE␣srcaddr='8.8.8.8'␣AND␣state=0␣AND␣prot=17␣AND␣
    srcport=53␣AND␣tcp_flags=0␣and␣dpkts<=2;"
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dns-request'␣WHERE␣dstaddr='8.8.8.8'␣AND␣state=0␣AND␣prot=17␣AND␣
    dstport=53␣AND␣tcp_flags=0␣and␣dpkts<=2;"

mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dns-response'␣WHERE␣srcaddr='8.8.4.4'␣AND␣state=0␣AND␣prot=17␣AND␣
    srcport=53␣AND␣tcp_flags=0␣and␣dpkts<=2;"
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dns-request'␣WHERE␣dstaddr='8.8.4.4'␣AND␣state=0␣AND␣prot=17␣AND␣
    dstport=53␣AND␣tcp_flags=0␣and␣dpkts<=2;"

mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dhcp-response'␣WHERE␣srcaddr='10.150.254.254'␣AND␣state=0␣AND␣prot=17␣
    AND␣srcport=67␣and␣dstport=68␣AND␣tcp_flags=0;"
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dhcp-request'␣WHERE␣dstaddr='10.150.254.254'␣AND␣state=0␣AND␣srcport=68
    ␣AND␣dstport=67␣AND␣prot=17";
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='dhcp-request-broadcast'␣WHERE␣dstaddr='255.255.255.255'␣AND␣state=0␣AND
    ␣srcport=68␣AND␣dstport=67␣AND␣prot=17";

mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='ntp'␣WHERE␣srcport=123␣and␣dstport=123␣and␣prot=17␣and␣tcp_flags=0";

mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',requesttype
    ='NetBIOS␣Name␣Service'␣WHERE␣srcport=137␣and␣dstport=137␣and␣prot=17␣and
    ␣tcp_flags=0";

mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣state='1',␣requesttype
    ='NetBIOS␣NetBIOS␣Datagram␣Service'␣WHERE␣srcport=138␣and␣dstport=138␣and
    ␣prot=17␣and␣tcp_flags=0";


mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣commet='debian8'␣WHERE
    ␣srcaddr='10.150.254.28'";
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣commet='debian7'␣WHERE
    ␣srcaddr='10.150.254.16'";
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣commet='debian7'␣WHERE
    ␣srcaddr='10.150.254.26'";
mysql -uthesis -pthesis flows -e "UPDATE␣flowinput␣SET␣commet='
```

```
    ubuntu14Desktop' WHERE srcaddr='10.150.254.17'";
32  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='ubuntu14Server
    ' WHERE srcaddr='10.150.254.29'";
33  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='ubuntu16Server
    ' WHERE srcaddr='10.150.254.18'";
34  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='ubuntu16Server
    ' WHERE srcaddr='10.150.254.30'";
35  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Windows7' 
    WHERE srcaddr='10.150.254.21'";
36  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Windows7' 
    WHERE srcaddr='10.150.254.32'";
37  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Windows10' 
    WHERE srcaddr='10.150.254.20'";
38  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Windows10' 
    WHERE srcaddr='10.150.254.31'";
39  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Mac OSX Sierra
    ' WHERE srcaddr='10.150.254.33'";
40  mysql -uthesis -pthesis flows -e "UPDATE flowinput SET commet='Open Suse 12' 
    WHERE srcaddr='10.150.254.34'";
```

Listing A.3: Flow-Caputure Konfiguration

```
1  # Configuration for flow-capture
2  -w /flowstore/data/ -d 128 -n 190 -N 3 -S 3 0/0/4432
```

Listing A.4: Softflow Deamon Konfiguration

```
1  # configuration for softflowd
2  # The interface softflowd listens on. You may also use "any" to listen on all
       interfaces.
3  INTERFACE="eth1"
4  OPTIONS="-n 127.0.0.1:4432"
```

B. Validierungsergebnisse der Betriebssystemdetektion

B.1. Bayes

B.1.1. BayesNet

Listing B.1: BayesNet

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES
    -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       14451               95.7273 %
Incorrectly Classified Instances       645                4.2727 %
Kappa statistic                          0.9397
Mean absolute error                      0.0103
Root mean squared error                  0.0872
Relative absolute error                  6.5372 %
Root relative squared error             31.1564 %
Total Number of Instances            15096
```

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,857    0,005    0,958      0,857    0,905      0,895    0,997     0,968     Mac OSX Sierra
                 0,913    0,001    0,986      0,913    0,948      0,946    0,997     0,968     ubuntu16Server
                 0,815    0,007    0,643      0,815    0,719      0,719    0,997     0,867     ubuntu14Desktop
                 0,973    0,008    0,988      0,973    0,980      0,967    0,997     0,997     Windows10
                 0,767    0,003    0,315      0,767    0,447      0,490    0,997     0,553     ubuntu14Server
                 0,971    0,017    0,830      0,971    0,895      0,888    0,992     0,973     Windows7
                 0,991    0,005    0,989      0,991    0,990      0,985    0,998     0,998     Open Suse 12
                 0,714    0,003    0,373      0,714    0,490      0,515    0,997     0,642     debian8
                 0,125    0,000    0,400      0,125    0,190      0,223    0,991     0,265     debian7
Weighted Avg.    0,957    0,007    0,963      0,957    0,959      0,950    0,997     0,986

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
 1513    0   80   41   12   97    8   15    0 |    a = Mac OSX Sierra
    0  717    1    7   10    3   33   14    0 |    b = ubuntu16Server
    3    3  189    4   22    4    4    3    0 |    c = ubuntu14Desktop
   30    7    1 6088    0  131    1    0    0 |    d = Windows10
    0    0    4    0   23    1    1    1    0 |    e = ubuntu14Server
   32    0    0    1    0 1179    1    0    1 |    f = Windows7
    1    0   18   17    5    3 4715    1    0 |    g = Open Suse 12
    0    0    1    0    0    3    4   25    2 |    h = debian8
    0    0    0    5    1    0    0    8    2 |    i = debian7
```

B.1.2. Naive Bayes

Listing B.2: Naive Bayes

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096

Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        7035               46.6017 %
Incorrectly Classified Instances      8061               53.3983 %
Kappa statistic                          0.3559
Mean absolute error                      0.1118
Root mean squared error                  0.3012
Relative absolute error                 71.3117 %
Root relative squared error            107.5723 %
Total Number of Instances            15096
```

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,009    0,005    0,190      0,009    0,017      0,017    0,968     0,776     Mac OSX Sierra
                 0,913    0,056    0,473      0,913    0,623      0,633    0,981     0,933     ubuntu16Server
                 0,034    0,001    0,471      0,034    0,064      0,124    0,953     0,354     ubuntu14Desktop
                 0,072    0,008    0,863      0,072    0,132      0,172    0,948     0,907     Windows10
                 0,133    0,001    0,308      0,133    0,186      0,202    0,956     0,136     ubuntu14Server
                 0,970    0,469    0,153      0,970    0,265      0,272    0,862     0,433     Windows7
                 0,980    0,058    0,887      0,980    0,931      0,899    0,984     0,988     Open Suse 12
                 0,000    0,000    0,000      0,000    0,000      -0,000   0,873     0,045     debian8
                 0,000    0,000    0,000      0,000    0,000      -0,000   0,904     0,015     debian7
Weighted Avg.    0,466    0,063    0,704      0,466    0,405      0,414    0,956     0,868

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
   16  677    4    8    4 1021   35    0    1 |    a = Mac OSX Sierra
    0  717    0    3    1   37   27    0    0 |    b = ubuntu16Server
    0   85    8    9    3  109   18    0    0 |    c = ubuntu14Desktop
   59   17    2  448    0 5258  473    1    0 |    d = Windows10
    0    0    0    2    4   11   13    0    0 |    e = ubuntu14Server
    6    8    0    3    1 1177   19    0    0 |    f = Windows7
    1   13    3   40    0   38 4665    0    0 |    g = Open Suse 12
    2    0    0    4    0   23    6    0    0 |    h = debian8
    0    0    0    2    0    9    5    0    0 |    i = debian7
```

B.1.3. Naive Bayes Multinomial Text

Listing B.3: Naive Bayes Multinomial Text

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayesMultinomialText -P 0 -M 3.0 -norm 1.0 -lnorm 2.0 -stopwords-
    handler weka.core.stopwords.Null -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t
    .,;:\\\'\\\"()?!\"" -stemmer weka.core.stemmers.NullStemmer
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096
Attributes:   10
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              os
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
Dictionary size: 0
The independent frequency of a class
--------------------------------------
Mac OSX Sierra  1767.0
ubuntu16Server  786.0
ubuntu14Desktop 233.0
Windows10       6259.0
ubuntu14Server  31.0
Windows7        1215.0
Open Suse 12    4761.0
debian8 36.0
debian7 17.0
```

```
The frequency of a word given the class
-----------------------------------------
Mac OSX Sie     ubuntu16Ser     ubuntu14Des       Windows10     ubuntu14Ser        Windows7     Open Suse 1
       debian8         debian7


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        6258               41.4547 %
Incorrectly Classified Instances      8838               58.5453 %
Kappa statistic                          0
Mean absolute error                      0.1568
Root mean squared error                  0.28
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            15096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,000    0,000    0,000      0,000    0,000      0,000    0,499     0,117     Mac OSX Sierra
                 0,000    0,000    0,000      0,000    0,000      0,000    0,498     0,052     ubuntu16Server
                 0,000    0,000    0,000      0,000    0,000      0,000    0,496     0,015     ubuntu14Desktop
                 1,000    1,000    0,415      1,000    0,586      0,000    0,500     0,414     Windows10
                 0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,002     ubuntu14Server
                 0,000    0,000    0,000      0,000    0,000      0,000    0,499     0,080     Windows7
                 0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,315     Open Suse 12
                 0,000    0,000    0,000      0,000    0,000      0,000    0,464     0,002     debian8
                 0,000    0,000    0,000      0,000    0,000      0,000    0,425     0,001     debian7
Weighted Avg.    0,415    0,415    0,172      0,415    0,243      0,000    0,499     0,294
```

```
=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
    0    0    0 1766    0    0    0    0    0 |    a = Mac OSX Sierra
    0    0    0  785    0    0    0    0    0 |    b = ubuntu16Server
    0    0    0  232    0    0    0    0    0 |    c = ubuntu14Desktop
    0    0    0 6258    0    0    0    0    0 |    d = Windows10
    0    0    0   30    0    0    0    0    0 |    e = ubuntu14Server
    0    0    0 1214    0    0    0    0    0 |    f = Windows7
    0    0    0 4760    0    0    0    0    0 |    g = Open Suse 12
    0    0    0   35    0    0    0    0    0 |    h = debian8
    0    0    0   16    0    0    0    0    0 |    i = debian7
```

B.2. Entscheidungsbäume

B.2.1. J48 Entscheidungsbaum

Listing B.4: J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096
Attributes:   10
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              os
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

Number of Leaves  :     8431

Size of the tree :      8735


Time taken to build model: 5.25 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       14233               94.2833 %
Incorrectly Classified Instances       863                5.7167 %
Kappa statistic                          0.9177
Mean absolute error                      0.0467
Root mean squared error                  0.1214
Relative absolute error                 29.796  %
Root relative squared error             43.3675 %
Total Number of Instances            15096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,817    0,002    0,985      0,817    0,893      0,885    0,994     0,966     Mac OSX Sierra
                 0,926    0,001    0,981      0,926    0,953      0,951    0,995     0,974     ubuntu16Server
                 0,642    0,001    0,925      0,642    0,758      0,768    0,961     0,816     ubuntu14Desktop
                 0,994    0,052    0,931      0,994    0,961      0,934    0,996     0,993     Windows10
                 0,167    0,000    0,500      0,167    0,250      0,288    0,821     0,197     ubuntu14Server
                 0,769    0,015    0,817      0,769    0,792      0,775    0,972     0,881     Windows7
                 0,996    0,013    0,973      0,996    0,984      0,977    0,999     0,998     Open Suse 12
                 0,371    0,000    0,765      0,371    0,500      0,532    0,966     0,667     debian8
                 0,250    0,000    0,571      0,250    0,348      0,378    0,913     0,502     debian7
Weighted Avg.    0,943    0,027    0,942      0,943    0,940      0,925    0,994     0,976

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
 1443    0    1  154    0  168    0    0    0 |    a = Mac OSX Sierra
    0  727    4   18    3    0   33    0    0 |    b = ubuntu16Server
    3    9  149   11    2    0   58    0    0 |    c = ubuntu14Desktop
    4    0    0 6220    0   32    2    0    0 |    d = Windows10
    0    5    5    0    5    1   14    0    0 |    e = ubuntu14Server
   15    0    0  263    0  933    3    0    0 |    f = Windows7
    0    0    2   11    0    8 4739    0    0 |    g = Open Suse 12
    0    0    0    2    0    0   17   13    3 |    h = debian8
    0    0    0    2    0    0    6    4    4 |    i = debian7
```

B.2.2. Konsolidierter J48 Entscheidungsbaum

Listing B.5: Konsolidierter J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48Consolidated -C 0.25 -M 2 -Q 1 -RM-C -RM-N 99.0 -RM-B -2 -RM-D 50.0
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096
Attributes:   10
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              os
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48Consolidated pruned tree
[RM] N_S=f(99% of coverage)=94 %Min=balanced Size=maxSize (without replacement)
 (*) Forced the 2-th class to be oversampled!!!
 (*) Forced the 4-th class to be oversampled!!!
 (*) Forced the 7-th class to be oversampled!!!
 (*) Forced the 8-th class to be oversampled!!!
True coverage achieved: 0.9956279724459038
------------------------------------------------------------------------------------------
Number of Leaves  :     859

Size of the tree :      891


Time taken to build model: 1.59 seconds

=== Stratified cross-validation ===
```

```
=== Summary ===

Correctly Classified Instances        8554               56.664  %
Incorrectly Classified Instances      6542               43.336  %
Kappa statistic                          0.455
Mean absolute error                      0.0967
Root mean squared error                  0.2219
Relative absolute error                 61.635  %
Root relative squared error             79.2344 %
Total Number of Instances            15096

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,580    0,108    0,416      0,580    0,485      0,411    0,850     0,659     Mac OSX Sierra
                 0,497    0,012    0,690      0,497    0,578      0,567    0,917     0,614     ubuntu16Server
                 0,759    0,164    0,068      0,759    0,124      0,194    0,903     0,672     ubuntu14Desktop
                 0,585    0,059    0,876      0,585    0,701      0,579    0,907     0,883     Windows10
                 0,200    0,002    0,207      0,200    0,203      0,202    0,744     0,102     ubuntu14Server
                 0,630    0,130    0,297      0,630    0,404      0,362    0,878     0,612     Windows7
                 0,528    0,001    0,995      0,528    0,690      0,656    0,895     0,831     Open Suse 12
                 0,429    0,000    0,789      0,429    0,556      0,581    0,799     0,341     debian8
                 0,250    0,009    0,030      0,250    0,053      0,083    0,767     0,207     debian7
Weighted Avg.    0,567    0,051    0,789      0,567    0,631      0,558    0,894     0,798

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
 1025   50  542   58    0   83    4    0    4 |    a = Mac OSX Sierra
    4  390  354    7   14   10    0    0    6 |    b = ubuntu16Server
    9    2  176   16    9   16    0    0    4 |    c = ubuntu14Desktop
  680   35  773 3659    0 1032    9    0   70 |    d = Windows10
    0    3    9    3    6    7    0    0    2 |    e = ubuntu14Server
  188    6  157   96    0  765    0    0    2 |    f = Windows7
  558   79  587  330    0  652 2514    0   40 |    g = Open Suse 12
    0    0    5    7    0    5    0   15    3 |    h = debian8
    0    0    4    1    0    3    0    4    4 |    i = debian7
```

B.2.3. Decision Stump Entscheidungsbaum

Listing B.6: Konsolidierter J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.DecisionStump
Relation:     QueryResult-weka.filters.unsupervised.attribute.Remove-R10
Instances:    15096
Attributes:   10
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              os
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Decision Stump

Classifications

duration <= 132.5 : Open Suse 12
duration > 132.5 : Windows10
duration is missing : Windows10
```

```
Class distributions

duration <= 132.5
a       b       c       d       e       f       g       h       i
0.1816  0.1051  0.0179  0.0732  0.0     0.0311  0.5910  0.0     0.0

duration > 132.5
a       b       c       d       e       f       g       h       i
0.0638  0.0082  0.0133  0.6956  0.0036  0.1210  0.0883  0.0042  0.0019

duration is missing
a       b       c       d       e       f       g       h       i
0.1170  0.0520  0.01546 0.4145  0.0020  0.0804  0.3153  0.0023  0.0011

a = Mac OSX Sierra              f = Windows7
b = ubuntu16Server              g = Open Suse 12
c = ubuntu14Desktop             h = debian8
d = Windows10                   i = debian7
e = ubuntu14Server

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        9787               64.8317 %
Incorrectly Classified Instances      5309               35.1683 %
Kappa statistic                          0.442
Mean absolute error                      0.1199
Root mean squared error                  0.2448
Relative absolute error                 76.4249 %
Root relative squared error             87.435  %
Total Number of Instances            15096
```

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,000    0,000    0,000      0,000    0,000      0,000    0,632     0,164     Mac OSX Sierra
                 0,000    0,000    0,000      0,000    0,000      0,000    0,732     0,098     ubuntu16Server
                 0,000    0,000    0,000      0,000    0,000      0,000    0,510     0,016     ubuntu14Desktop
                 0,920    0,285    0,696      0,920    0,792      0,629    0,811     0,671     Windows10
                 0,000    0,000    0,000      0,000    0,000      0,000    0,724     0,004     ubuntu14Server
                 0,000    0,000    0,000      0,000    0,000      0,000    0,645     0,114     Windows7
                 0,846    0,270    0,591      0,846    0,696      0,538    0,783     0,557     Open Suse 12
                 0,000    0,000    0,000      0,000    0,000      0,000    0,707     0,004     debian8
                 0,000    0,000    0,000      0,000    0,000      0,000    0,683     0,002     debian7
Weighted Avg.    0,648    0,203    0,475      0,648    0,548      0,430    0,759     0,487

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i   <-- classified as
    0    0    0  528    0    0 1238    0    0 |    a = Mac OSX Sierra
    0    0    0   68    0    0  717    0    0 |    b = ubuntu16Server
    0    0    0  110    0    0  122    0    0 |    c = ubuntu14Desktop
    0    0    0 5758    0    0  500    0    0 |    d = Windows10
    0    0    0   30    0    0    0    0    0 |    e = ubuntu14Server
    0    0    0 1002    0    0  212    0    0 |    f = Windows7
    0    0    0  731    0    0 4029    0    0 |    g = Open Suse 12
    0    0    0   35    0    0    0    0    0 |    h = debian8
    0    0    0   16    0    0    0    0    0 |    i = debian7
```

C. Validierungsergebnisse der Softwareklassifizierung

C.1. Bayes

C.1.1. BayesNet

Auf Basis der Testdaten war keine Klassifikation mit BayesNet möglich.

C.1.2. Naive Bayes

Listing C.1: Naive Bayes

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     QueryResult
Instances:    525
Attributes:   11
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              requesttype
              software
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
```

```
Naive Bayes Classifier


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         263               50.0952 %
Incorrectly Classified Instances       262               49.9048 %
Kappa statistic                          0.2737
Mean absolute error                      0.1856
Root mean squared error                  0.3503
Relative absolute error                103.1506 %
Root relative squared error            117.204  %
Total Number of Instances              525

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,269    0,018    0,837      0,269    0,407      0,399    0,833     0,660     Dropbox
                 0,500    0,012    0,400      0,500    0,444      0,438    0,811     0,383     SSH
                 0,478    0,066    0,250      0,478    0,328      0,305    0,720     0,271     TeamViewer
                 0,594    0,179    0,848      0,594    0,699      0,403    0,777     0,825     Skype
                 0,000    0,004    0,000      0,000    0,000      -0,010   0,623     0,041     OpenOffice
                 1,000    0,352    0,082      1,000    0,152      0,231    0,991     0,615     MySQL
Weighted Avg.    0,501    0,131    0,767      0,501    0,569      0,382    0,792     0,725

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
  36   5  16  28   2  47 |   a = Dropbox
   0   4   0   0   0   4 |   b = SSH
   1   0  11   5   0   6 |   c = TeamViewer
   4   1  16 196   0 113 |   d = Skype
   2   0   1   2   0   9 |   e = OpenOffice
   0   0   0   0   0  16 |   f = MySQL
```

C.1.3. Naive Bayes Multinomial Text

Listing C.2: Naive Bayes Multinomial Text

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayesMultinomialText -P 0 -M 3.0 -norm 1.0 -lnorm 2.0 -stopwords-
    handler weka.core.stopwords.Null -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t
    .,;:\\\'\\\"()?!\"" -stemmer weka.core.stemmers.NullStemmer
Relation:     QueryResult
Instances:    525
Attributes:   11
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              requesttype
              software
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Dictionary size: 0

The independent frequency of a class
--------------------------------------
Dropbox 135.0
SSH     9.0
TeamViewer      24.0
Skype   331.0
OpenOffice      15.0
MySQL   17.0
```

```
The frequency of a word given the class
-----------------------------------------
    Dropbox             SSH      TeamViewer           Skype      OpenOffice           MySQL


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         330               62.8571 %
Incorrectly Classified Instances       195               37.1429 %
Kappa statistic                          0
Mean absolute error                      0.1799
Root mean squared error                  0.2989
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              525

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,000    0,000    0,000      0,000    0,000      0,000    0,486     0,249     Dropbox
                 0,000    0,000    0,000      0,000    0,000      0,000    0,398     0,014     SSH
                 0,000    0,000    0,000      0,000    0,000      0,000    0,453     0,040     TeamViewer
                 1,000    1,000    0,629      1,000    0,772      0,000    0,494     0,626     Skype
                 0,000    0,000    0,000      0,000    0,000      0,000    0,414     0,023     OpenOffice
                 0,000    0,000    0,000      0,000    0,000      0,000    0,420     0,027     MySQL
Weighted Avg.    0,629    0,629    0,395      0,629    0,485      0,000    0,484     0,460

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   0   0 134   0   0 |   a = Dropbox
   0   0   0   8   0   0 |   b = SSH
   0   0   0  23   0   0 |   c = TeamViewer
   0   0   0 330   0   0 |   d = Skype
   0   0   0  14   0   0 |   e = OpenOffice
   0   0   0  16   0   0 |   f = MySQL
```

C.2. Entscheidungsbäume

C.2.1. J48 Entscheidungsbaum

Listing C.3: J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     QueryResult
Instances:    525
Attributes:   11
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              requesttype
              software
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

srcport <= 51903: Skype (470.0/153.0)
srcport > 51903: Dropbox (55.0/36.0)

Number of Leaves  :     2

Size of the tree :      3


Time taken to build model: 0.06 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         339               64.5714 %
Incorrectly Classified Instances       186               35.4286 %
Kappa statistic                          0.1937
Mean absolute error                      0.1614
Root mean squared error                  0.2874
Relative absolute error                 89.7117 %
Root relative squared error             96.1578 %
Total Number of Instances              525

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0,291     0,120     0,453       0,291   0,355      0,201    0,671     0,409     Dropbox
                 0,000     0,000     0,000       0,000   0,000      0,000    0,822     0,087     SSH
                 0,000     0,000     0,000       0,000   0,000      0,000    0,671     0,111     TeamViewer
                 0,903     0,692     0,688       0,903   0,781      0,268    0,664     0,737     Skype
                 0,000     0,000     0,000       0,000   0,000      0,000    0,707     0,057     OpenOffice
                 0,125     0,008     0,333       0,125   0,182      0,189    0,843     0,127     MySQL
Weighted Avg.    0,646     0,466     0,559       0,646   0,587      0,225    0,675     0,579

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
  39   0   0  93   0   2 |   a = Dropbox
   3   0   0   4   0   1 |   b = SSH
   6   0   0  17   0   0 |   c = TeamViewer
  31   0   0 298   0   1 |   d = Skype
   4   0   0  10   0   0 |   e = OpenOffice
   3   0   0  11   0   2 |   f = MySQL
```

C.2.2. Konsolidierter J48 Entscheidungsbaum

Listing C.4: Konsolidierter J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48Consolidated -C 0.25 -M 2 -Q 1 -RM-C -RM-N 99.0 -RM-B -2 -RM-D 50.0
Relation:     QueryResult
Instances:    525
Attributes:   11
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              requesttype
              software
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48Consolidated pruned tree
[RM] N_S=f(99% of coverage)=136 %Min=balanced Size=maxSize (without replacement)
 (*) Forced the 1-th class to be oversampled!!!
True coverage achieved: 0.9937816982282264
------------------------------------------------------------------------------------------------------

Number of Leaves  :     137

Size of the tree :      139


Time taken to build model: 0.07 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         174               33.1429 %
Incorrectly Classified Instances       351               66.8571 %
Kappa statistic                          0.1875
Mean absolute error                      0.2047
Root mean squared error                  0.3482
Relative absolute error                113.7923 %
Root relative squared error            116.4826 %
Total Number of Instances              525

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,381    0,095    0,580      0,381    0,459      0,334    0,730     0,533     Dropbox
                 0,625    0,230    0,040      0,625    0,076      0,114    0,754     0,504     SSH
                 0,348    0,090    0,151      0,348    0,211      0,175    0,672     0,091     TeamViewer
                 0,288    0,015    0,969      0,288    0,444      0,338    0,772     0,846     Skype
                 0,143    0,106    0,036      0,143    0,057      0,019    0,518     0,060     OpenOffice
                 0,813    0,183    0,123      0,813    0,213      0,270    0,911     0,398     MySQL
Weighted Avg.    0,331    0,050    0,769      0,331    0,415      0,316    0,754     0,693

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 51 35 16  3 18 11 |  a = Dropbox
  0  5  0  0  1  2 |  b = SSH
  3  4  8  0  0  8 |  c = TeamViewer
 29 75 29 95 32 70 |  d = Skype
  5  5  0  0  2  2 |  e = OpenOffice
  0  0  0  0  3 13 |  f = MySQL
```

C.2.3. Best First Entscheidungsbaum

Listing C.5: Konsolidierter J48 Entscheidungsbaum

```
=== Run information ===

Scheme:       weka.classifiers.trees.BFTree -M 2 -N 5 -C 1.0 -P POSTPRUNED -S 1
Relation:     QueryResult
Instances:    525
Attributes:   11
              dpkts
              doctets
              duration
              srcaddr
              dstaddr
              srcport
              dstport
              prot
              tcp_flags
              requesttype
              software
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Best-First Decision Tree

Size of the Tree: 77

Number of Leaf Nodes: 39

Time taken to build model: 1.87 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         467               88.9524 %
Incorrectly Classified Instances        58               11.0476 %
Kappa statistic                          0.7859
Mean absolute error                      0.0454
Root mean squared error                  0.1782
Relative absolute error                 25.2083 %
Root relative squared error             59.6132 %
Total Number of Instances              525

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,881    0,059    0,837      0,881    0,858      0,808    0,938     0,842     Dropbox
                 0,875    0,000    1,000      0,875    0,933      0,935    1,000     1,000     SSH
                 0,087    0,016    0,200      0,087    0,121      0,106    0,696     0,142     TeamViewer
                 0,967    0,133    0,925      0,967    0,945      0,848    0,944     0,948     Skype
                 0,357    0,000    1,000      0,357    0,526      0,592    0,796     0,406     OpenOffice
                 1,000    0,002    0,941      1,000    0,970      0,969    1,000     1,000     MySQL
Weighted Avg.    0,890    0,100    0,874      0,890    0,876      0,804    0,930     0,874

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
 118   0   4  12   0   0 |   a = Dropbox
   0   7   0   0   0   1 |   b = SSH
   9   0   2  12   0   0 |   c = TeamViewer
   7   0   4 319   0   0 |   d = Skype
   7   0   0   2   5   0 |   e = OpenOffice
   0   0   0   0   0  16 |   f = MySQL
```

D. Gesetzesauszüge

Nachfolgend werden in dieser Arbeit genutzte Gesetzestexte mit Stand vom 15.11.2016 auszugsweise abgedruckt.

D.1. Strafgesetzbuch

D.1.1. § 202 StGB Verletzung des Briefgeheimnisses

(1) Wer unbefugt
1. einen verschlossenen Brief oder ein anderes verschlossenes Schriftstück, die nicht zu seiner Kenntnis bestimmt sind, öffnet oder
2. sich vom Inhalt eines solchen Schriftstücks ohne Öffnung des Verschlusses unter Anwendung technischer Mittel Kenntnis verschafft, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft, wenn die Tat nicht in § 206 mit Strafe bedroht ist.

(2) Ebenso wird bestraft, wer sich unbefugt vom Inhalt eines Schriftstücks, das nicht zu seiner Kenntnis bestimmt und durch ein verschlossenes Behältnis gegen Kenntnisnahme besonders gesichert ist, Kenntnis verschafft, nachdem er dazu das Behältnis geöffnet hat.

(3) Einem Schriftstück im Sinne der Absätze 1 und 2 steht eine Abbildung gleich.

D.1.2. § 202a Ausspähen von Daten

(1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

D.1.3. § 202b Abfangen von Daten

Wer unbefugt sich oder einem anderen unter Anwendung von technischen Mitteln nicht für ihn bestimmte Daten (§ 202a Abs. 2) aus einer nichtöffentlichen Datenübermittlung oder aus der elektromagnetischen Abstrahlung einer Datenverarbeitungsanlage verschafft, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft, wenn die Tat nicht in anderen Vorschriften mit schwererer Strafe bedroht ist.

D.1.4. § 202c Vorbereiten des Ausspähens und Abfangens von Daten

(1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er

1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist, herstellt, sich oder

einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.

(2) § 149 Abs. 2 und 3 gilt entsprechend.

D.2. Bundesdatenschutzgesetz

D.2.1. § 3a Datenvermeidung und Datensparsamkeit

Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten und die Auswahl und Gestaltung von Datenverarbeitungssystemen sind an dem Ziel auszurichten, so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen. Insbesondere sind personenbezogene Daten zu anonymisieren oder zu pseudonymisieren, soweit dies nach dem Verwendungszweck möglich ist und keinen im Verhältnis zu dem angestrebten Schutzzweck unverhältnismäßigen Aufwand erfordert.

D.2.2. § 4 Zulässigkeit der Datenerhebung, -verarbeitung und -nutzung

(1) Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten sind nur zulässig, soweit dieses Gesetz oder eine andere Rechtsvorschrift dies erlaubt oder anordnet oder der Betroffene eingewilligt hat.

(2) Personenbezogene Daten sind beim Betroffenen zu erheben. Ohne seine Mitwirkung dürfen sie nur erhoben werden, wenn

1. eine Rechtsvorschrift dies vorsieht oder zwingend voraussetzt oder

2.

a) die zu erfüllende Verwaltungsaufgabe ihrer Art nach oder der Geschäftszweck eine Erhebung bei anderen Personen oder Stellen erforderlich macht oder

b) die Erhebung beim Betroffenen einen unverhältnismäßigen Aufwand erfordern würde

und keine Anhaltspunkte dafür bestehen, dass überwiegende schutzwürdige Interessen des Betroffenen beeinträchtigt werden.

(3) Werden personenbezogene Daten beim Betroffenen erhoben, so ist er, sofern er nicht bereits auf andere Weise Kenntnis erlangt hat, von der verantwortlichen Stelle über

1. die Identität der verantwortlichen Stelle,

2. die Zweckbestimmungen der Erhebung, Verarbeitung oder Nutzung und

3. die Kategorien von Empfängern nur, soweit der Betroffene nach den Umständen des Einzelfalles nicht mit der Übermittlung an diese rechnen muss,

zu unterrichten. Werden personenbezogene Daten beim Betroffenen aufgrund einer Rechtsvorschrift erhoben, die zur Auskunft verpflichtet, oder ist die Erteilung der Auskunft Voraussetzung für die Gewährung von Rechtsvorteilen, so ist der Betroffene hierauf, sonst auf die Freiwilligkeit seiner Angaben hinzuweisen. Soweit nach den Umständen des Einzelfalles erforderlich oder auf Verlangen, ist er über die Rechtsvorschrift und über die Folgen der Verweigerung von Angaben aufzuklären.

D.2.3. § 9 Technische und organisatorische Maßnahmen

Öffentliche und nicht-öffentliche Stellen, die selbst oder im Auftrag personenbezogene Daten erheben, verarbeiten oder nutzen, haben die technischen und organisatorischen Maßnahmen

zu treffen, die erforderlich sind, um die Ausführung der Vorschriften dieses Gesetzes, insbesondere die in der Anlage zu diesem Gesetz genannten Anforderungen, zu gewährleisten. Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht.

D.2.4. § 32 Datenerhebung, -verarbeitung und -nutzung für Zwecke des Beschäftigungsverhältnisses

(1) Personenbezogene Daten eines Beschäftigten dürfen für Zwecke des Beschäftigungsverhältnisses erhoben, verarbeitet oder genutzt werden, wenn dies für die Entscheidung über die Begründung eines Beschäftigungsverhältnisses oder nach Begründung des Beschäftigungsverhältnisses für dessen Durchführung oder Beendigung erforderlich ist. Zur Aufdeckung von Straftaten dürfen personenbezogene Daten eines Beschäftigten nur dann erhoben, verarbeitet oder genutzt werden, wenn zu dokumentierende tatsächliche Anhaltspunkte den Verdacht begründen, dass der Betroffene im Beschäftigungsverhältnis eine Straftat begangen hat, die Erhebung, Verarbeitung oder Nutzung zur Aufdeckung erforderlich ist und das schutzwürdige Interesse des Beschäftigten an dem Ausschluss der Erhebung, Verarbeitung oder Nutzung nicht überwiegt, insbesondere Art und Ausmaß im Hinblick auf den Anlass nicht unverhältnismäßig sind.

(2) Absatz 1 ist auch anzuwenden, wenn personenbezogene Daten erhoben, verarbeitet oder genutzt werden, ohne dass sie automatisiert verarbeitet oder in oder aus einer nicht automatisierten Datei verarbeitet, genutzt oder für die Verarbeitung oder Nutzung in einer solchen Datei erhoben werden.

(3) Die Beteiligungsrechte der Interessenvertretungen der Beschäftigten bleiben unberührt.

D.3. Bayerisches Datenschutzgesetz

D.3.1. Art. 15 Zulässigkeit der Datenerhebung, -verarbeitung und -nutzung

(1) Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten sind nur zulässig, wenn
1.dieses Gesetz oder eine andere Rechtsvorschrift sie erlaubt oder anordnet oder
2.der Betroffene eingewilligt hat.
(2) Wird eine Einwilligung eingeholt, so sind Betroffene auf den Zweck der Erhebung, Verarbeitung oder Nutzung, auf die Empfänger vorgesehener Übermittlungen sowie unter Darlegung der Rechtsfolgen darauf hinzuweisen, dass sie die Einwilligung verweigern können.
(3) 1Die Einwilligung ist schriftlich oder elektronisch zu erteilen, soweit nicht wegen besonderer Umstände eine andere Form angemessen ist.2Im Bereich der wissenschaftlichen Forschung liegen solche besonderen Umstände auch dann vor, wenn der bestimmte Forschungszweck durch die schriftliche oder elektronische Einwilligung erheblich beeinträchtigt würde.3In diesem Fall sind der Hinweis gemäß Absatz 2 und die Gründe, aus denen sich die erhebliche Beeinträchtigung des wissenschaftlichen Forschungszwecks ergibt, festzuhalten.
(4) Soll die Einwilligung zusammen mit anderen Erklärungen schriftlich erteilt werden, ist die Einwilligungserklärung im äußeren Erscheinungsbild der Erklärung hervorzuheben.2Bei elektronischer Einwilligung ist sicherzustellen, dass
1. der Betroffene die Einwilligung bewusst und eindeutig erteilt hat,
2. er sich über ihren Inhalt nachträglich informieren und sie mit Wirkung für die Zukunft

widerrufen kann und
3. die Einwilligung protokolliert wird.
(5) Widersprechen Betroffene einer bestimmten Erhebung, Verarbeitung oder Nutzung und ergibt eine Abwägung im Einzelfall, dass das schutzwürdige Interesse eines Betroffenen wegen seiner besonderen persönlichen Situation das Interesse der öffentlichen Stelle an der Erhebung, Verarbeitung oder Nutzung dieser Daten überwiegt, so dürfen insoweit personenbezogene Daten nicht erhoben, verarbeitet oder genutzt werden.2Satz 1 gilt nicht, wenn eine Rechtsvorschrift die Erhebung, Verarbeitung oder Nutzung anordnet.
(6) Entscheidungen, die für Betroffene eine rechtliche Folge nach sich ziehen oder sie erheblich beeinträchtigen, dürfen nicht ausschließlich auf eine automatisierte Verarbeitung oder Nutzung zum Zweck der Bewertung einzelner Persönlichkeitsmerkmale gestützt werden.
Satz 1 gilt nicht, soweit
1. eine Rechtsvorschrift dies ausdrücklich vorsieht,
2. damit dem Begehren der Betroffenen stattgegeben wird, oder
3. den Betroffenen die Tatsache einer Entscheidung nach Satz 1 mitgeteilt wird und ihnen Gelegenheit gegeben wird, ihren Standpunkt geltend zu machen; die öffentliche Stelle ist verpflichtet, nach Eingang der Stellungnahme ihre Entscheidung erneut zu prüfen.
(7) Das Erheben, Verarbeiten oder Nutzen personenbezogener Daten, aus denen die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen oder die Gewerkschaftszugehörigkeit hervorgehen, sowie von Daten über Gesundheit oder Sexualleben, ist über die Vorschriften dieses Abschnitts hinaus nur zulässig, wenn
1. eine Rechtsvorschrift dies ausdrücklich vorsieht,
2. die Betroffenen eingewilligt haben, wobei sich die Einwilligung ausdrücklich auf diese Daten beziehen muss,
3. es zum Schutz lebenswichtiger Interessen Betroffener oder Dritter erforderlich ist, sofern die Betroffenen aus physischen oder rechtlichen Gründen außerstande sind, ihre Einwilligung zu geben,
4. es sich um Daten handelt, die Betroffene offenkundig öffentlich gemacht haben,
5. es zur Abwehr erheblicher Nachteile für das Gemeinwohl oder von Gefahren für die öffentliche Sicherheit und Ordnung erforderlich ist,
6. es zur Verfolgung von Straftaten oder Ordnungswidrigkeiten, zur Vollstreckung oder zum Vollzug von Strafen oder Maßnahmen im Sinn des § 11 Abs. 1 Nr. 8 des Strafgesetzbuchs oder von Erziehungsmaßregeln oder Zuchtmitteln im Sinn des Jugendgerichtsgesetzes oder zur Vollstreckung von Bußgeldentscheidungen erforderlich ist,
7. es zur Durchführung wissenschaftlicher Forschung erforderlich ist, das wissenschaftliche Interesse an der Durchführung des Forschungsvorhabens das Interesse des Betroffenen an dem Ausschluss der Erhebung, Verarbeitung oder Nutzung erheblich überwiegt und der Zweck der Forschung auf andere Weise nicht oder nur mit unverhältnismäßigem Aufwand erreicht werden kann,
8. es erforderlich ist, um den Rechten und Pflichten der öffentlichen Stellen auf dem Gebiet des Dienst- und Arbeitsrechts Rechnung zu tragen, oder
9. es zum Zweck der Gesundheitsvorsorge, der medizinischen Diagnostik, der Gesundheitsversorgung oder Behandlung oder für die Verwaltung von Gesundheitsdiensten erforderlich ist und die Verarbeitung dieser Daten durch ärztliches Personal oder durch sonstige Personen erfolgt, die einer entsprechenden Geheimhaltungspflicht unterliegen.
2 Art. 20 bleibt unberührt.
(8) Die Absätze 5 bis 7 gelten für Strafgerichte nur, soweit sie in Verwaltungsangelegen-

heiten tätig werden. Die Absätze 5 bis 7 gelten nicht für Behörden der Staatsanwaltschaft, für Justizvollzugsanstalten, für Führungsaufsichtsstellen und für Stellen der Gerichts- und Bewährungshilfe.

D.3.2. Art. 16 Erhebung

(1) Das Erheben personenbezogener Daten ist zulässig, wenn ihre Kenntnis zur Erfüllung der in der Zuständigkeit der erhebenden Stelle liegenden Aufgaben erforderlich ist.
(2) 1Personenbezogene Daten, die nicht aus allgemein zugänglichen Quellen entnommen werden, sind beim Betroffenen mit seiner Kenntnis zu erheben.2Personenbezogene Daten dürfen bei Dritten nur erhoben werden, wenn
1. eine Rechtsvorschrift eine solche Erhebung vorsieht oder zwingend voraussetzt,
2.
a) die zu erfüllende Verwaltungsaufgabe ihrer Art nach oder im Einzelfall eine solche Erhebung erforderlich macht oder
b) die Erhebung beim Betroffenen einen unverhältnismäßigen Aufwand erfordern würde oder keinen Erfolg verspricht und keine Anhaltspunkte dafür bestehen, daß überwiegende schutzwürdige Interessen des Betroffenen beeinträchtigt werden, oder
3. die Daten nach Art. 18 Abs. 1 oder einer anderen Rechtsvorschrift von einer öffentlichen Stelle an die erhebende Stelle übermittelt werden dürfen. 3Werden Daten beim Betroffenen ohne seine Kenntnis erhoben, gelten die Nummern 1 und 2 Buchst. a des Satzes 2 entsprechend.
(3) Werden personenbezogene Daten beim Betroffenen mit seiner Kenntnis erhoben, so ist der Erhebungszweck ihm gegenüber anzugeben.2Werden sie beim Betroffenen auf Grund einer Rechtsvorschrift erhoben, die zur Auskunft verpflichtet, oder ist die Erteilung der Auskunft Voraussetzung für die Gewährung von Rechtsvorteilen, so ist der Betroffene hierauf, sonst auf die Freiwilligkeit seiner Angaben hinzuweisen.3Auf Verlangen ist der Betroffene über die Rechtsvorschrift und über die Folgen der Verweigerung von Angaben aufzuklären.4Bei einer Datenerhebung auf schriftlichem Weg ist die Rechtsvorschrift stets anzugeben.
(4) Werden personenbezogene Daten statt beim Betroffenen bei einer nicht-öffentlichen Stelle erhoben, so ist die Stelle auf die Rechtsvorschrift, die zur Auskunft verpflichtet, sonst auf die Freiwilligkeit ihrer Angaben hinzuweisen.

D.3.3. Art. 17 Verarbeitung und Nutzung

(1) Das Speichern, Verändern oder Nutzen personenbezogener Daten ist zulässig, wenn
1. es zur Erfüllung der in der Zuständigkeit der speichernden Stelle liegenden Aufgaben erforderlich ist und
2. es für die Zwecke erfolgt, für die die Daten erhoben worden sind; ist keine Erhebung vorausgegangen, dürfen die Daten nur für die Zwecke geändert oder genutzt werden, für die sie gespeichert worden sind.
(2) Abweichend von Absatz 1 Nr. 2 ist das Speichern, Verändern oder Nutzen personenbezogener Daten für andere Zwecke zulässig, wenn
1. eine Rechtsvorschrift dies vorsieht oder zwingend voraussetzt oder die Beteiligung von Trägern öffentlicher Belange bestimmt,
2. der Betroffene eingewilligt hat,

3. offensichtlich ist, daß es im Interesse des Betroffenen liegt, und kein Grund zu der Annahme besteht, daß er in Kenntnis des anderen Zwecks seine Einwilligung hierzu verweigern würde,
4. die Daten für den anderen Zweck auf Grund einer durch Rechtsvorschrift festgelegten Auskunfts- oder Meldepflicht beim Betroffenen erhoben werden dürfen und der Betroffene dieser Pflicht nicht nachgekommen ist,
5. Angaben des Betroffenen überprüft werden sollen, weil tatsächliche Anhaltspunkte für deren Unrichtigkeit bestehen,
6. Angaben des Betroffenen zur Erlangung von finanziellen Leistungen öffentlicher Stellen mit anderen derartigen Angaben verglichen werden sollen,
7. es zur Entscheidung über die Verleihung von staatlichen Orden oder Ehrenzeichen oder von sonstigen staatlichen Ehrungen erforderlich ist,
8. die Daten aus allgemein zugänglichen Quellen entnommen werden können oder die speichernde Stelle die Daten veröffentlichen dürfte,
9. es zur Abwehr erheblicher Nachteile für das Gemeinwohl oder von Gefahren für die öffentliche Sicherheit oder Ordnung oder zur Abwehr einer schwerwiegenden Beeinträchtigung der Rechte einer anderen Person erforderlich ist,
10. es zur Verfolgung von Straftaten oder Ordnungswidrigkeiten, zur Vollstreckung oder zum Vollzug von Strafen oder Maßnahmen im Sinn des § 11 Abs. 1 Nr. 8 des Strafgesetzbuchs oder von Erziehungsmaßregeln oder Zuchtmitteln im Sinn des Jugendgerichtsgesetzes oder zur Vollstreckung von Bußgeldentscheidungen erforderlich ist oder
11. es zur Durchführung wissenschaftlicher Forschung erforderlich ist, das wissenschaftliche Interesse an der Durchführung des Forschungsvorhabens das Interesse des Betroffenen an dem Ausschluß der Zweckänderung erheblich überwiegt und der Zweck der Forschung auf andere Weise nicht oder nur mit unverhältnismäßigem Aufwand erreicht werden kann.
(3) Eine Verarbeitung oder Nutzung für andere Zwecke liegt nicht vor, wenn sie der Wahrnehmung von Aufsichts- oder Kontrollbefugnissen, der Erstellung von Geschäftsstatistiken, der Rechnungsprüfung, der Durchführung von Organisationsuntersuchungen für die speichernde Stelle oder der Prüfung oder Wartung automatisierter Verfahren der Datenverarbeitung dient.2Das gilt auch für die Verarbeitung und Nutzung zu Ausbildungs- oder Prüfungszwecken durch die speichernde Stelle, soweit nicht offensichtlich überwiegende schutzwürdige Interessen des Betroffenen entgegenstehen.
(4) Personenbezogene Daten in automatisierten Dateien im Sinn des Art. 2 Abs. 3 sowie personenbezogene Daten, die ausschließlich zu Zwecken der Datenschutzkontrolle, der Datensicherung oder zur Sicherstellung eines ordnungsgemäßen Betriebes einer Datenverarbeitungsanlage gespeichert werden, dürfen nur für diese Zwecke verarbeitet oder genutzt werden.
(5) Sind mit personenbezogenen Daten, die nach den Absätzen 1 bis 3 durch Weitergabe innerhalb der speichernden Stelle genutzt werden dürfen, weitere personenbezogene Daten des Betroffenen oder Dritter in Akten so verbunden, daß eine Trennung nicht oder nur mit unvertretbarem Aufwand möglich ist, so ist die Weitergabe auch dieser Daten zulässig, soweit nicht offensichtlich überwiegende schutzwürdige Interessen des Betroffenen oder Dritter entgegenstehen.2Eine darüber hinausgehende Nutzung oder Verarbeitung dieser Daten ist nur zulässig, soweit die Daten auch hierfür hätten weitergegeben werden dürfen.

Abkürzungsverzeichnis

API	application programming interface
BAdW	Bayerische Akademie der Wissenschaften
BayDSG	Bayerisches Datenschutzgesetz
BDSG	Bundesdatenschutzgesetz
CDN	Content Delivery Network
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DPI	Deep Packet Inspection
Dr. Portscan	Delta Reporting Portscan
FRF-Tool	Flow-Record-Fingerprinting-Tool
HTTP	Hypertext Transfer Protocol
HM	Hochschule München
ICMP	Internet Control Message Protocol
IfI	Institut für Informatik
IIS	Internet Information Services
IP	Interntprotokoll
ITAM	IT-Asset-Management
ITIL	IT Infrastructure Library
LMU	Ludwig-Maximilians-Universität
LRZ	Leibniz-Rechenzentrum
MWN	Münchner Wissenschaftsnetz
MTU	Maximum Transmission Unit
NAT	Network Address Translation
Nmap	Network Mapper

OS	Operation System
PRADS	Passive Real-time Asset Detection System
SNMP	Simple Network Management Protocol
SSH	Secure Shell
StWM	Studentenwerk München
SYN	synchronize (Netzwerk)
TCP	Transmission Control Protocol
TKÜ	Telekommunikationsüberwachung
TRTKÜV	Technische Richtlinie zur Umsetzung gesetzlicher Maßnahmen zur Überwachung der Telekommunikation und zum Auskunftersuchen für Verkehrsdaten
TTL	Time to Live
TUM	Technische Universität München
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VM	Virtuelle Maschine
VPN	Virtual Private Network
WWW	World Wide Web

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Ark02] Arkin, Ofir: *A remote active OS fingerprinting tool using ICMP*. login: the Magazine of USENIX and Sage, 27(2):14–19, 2002.

[AXE11] AXELOS Limited: *ITIL Glossary of Terms English v.1.0*, 2011.

[AY02] Arkin, Ofir und Fyodor Yarochkin: *A "Fuzzy" Approach to Remote Active Operating System Fingerprinting*, 2002.

[Bed09] Bedner, Mark: *Rechtmäßigkeit der 'Deep Packet Inspection'*, 2009.

[Ber14] Bernhard, Andreas: *Netzbasierte Erkennung von Systemen und Diensten zur Verbesserung der IT–Sicherheit*. Bachelorarbeit, Ludwig-Maximilians-Universität München, München, März 2014.

[BK11] Barg, Alexander und Gregory Kabatiansky: *Fingerprinting*. Encyclopedia of Cryptography and Security, 2:465–467, 2011.

[BND] BND: *SIGINT*. Besucht am 07.06.2016 `http://www.bnd.bund.de/DE/Auftrag/Informationsgewinnung/SIGINT/sigint_node.html`.

[Bun] Bundestag: *Die Arbeit der Nachrichtendienste*. Besucht am 07.06.2016 `https://www.bundestag.de/bundestag/gremien18/pkgr/nachrichtendienste/248040`.

[Bun14] Bundesamt für Verfassungsschutz: *Spionage - Ihre Ziele - Ihre Methoden*, 2014. Besucht am 16. Mai 2016 `https://www.verfassungsschutz.de/de/oeffentlichkeitsarbeit/publikationen/pb-spionage-und-proliferationsabwehr/broschuere-2014-05-spionage-ihre-ziele-ihre-methoden`.

[Cap90] Caplan, Richard M.: *How fingerprints came into use for personal identification*. Journal of the American Academy of Dermatology, 23:109–114, 1990.

[Chr] Christian Lehmann: *Mengen und Klassen*. Besucht am 20.09.2016 `http://www.christianlehmann.eu/ling/epistemology/concepts/klassifikation.php`.

[Die08] Dietzfelbinger, Martin: *Fingerprinting*. In: Voecking, Berthold, Helmut Alt, Martin Dietzfelbinger, Ruediger Reischuk, Christian Scheideler, Heribert Vollmer und Dorothea Wagner (Herausgeber): *Taschenbuch der Algorithmen*, eXamen.press, Seiten 193–204. Springer Berlin Heidelberg, 2008.

[Dir02] Dirk Loss: *Data Mining: Klassifikations- und Clusteringverfahren*, April 2002.

[Gab16] GABLER WIRTSCHAFTSLEXIKON: *IT-Management*, Januar 2016. Besucht am 15.05.2016 `http://wirtschaftslexikon.gabler.de/Definition/it-management.html`.

[gar] GARTNER: *IT Asset Management (ITAM)*. Besucht am 15.05.2016 `http://www.gartner.com/it-glossary/it-asset-management-itam/`.

[GvE05] GRABATIN, MICHAEL und FELIX VON EYE: *Size Matters*. Linux Magazin, Februar 2005. `http://www.linux-magazin.de/Ausgaben/2016/02/Metadatenanalyse`.

[GvE16] GRABATIN, MICHAEL und FELIX VON EYE: *Passive Detektion von Betriebssystem und installierter Software mittels Flow-Records*. In: PAULSEN, CHRISTIAN (Herausgeber): *Sicherheit in vernetzten Systemen: 23. DFN-Konferenz*, Seiten A–1–A–16, Norderstedt, Deutschland, Februar 2016. Books on Demand.

[Han] HANS LOHNINGER: *Kreuzvalidierung*. Besucht am 11.10.2016 `http://www.statistics4u.info/fundstat_germ/cc_cross_validation.html`.

[HR15] HOMMEL, WOLFGANG und HELMUT REISER: *Das Münchner Wissenschaftsnetz (MWN) Konzepte, Dienste, Infrastruktur und Management*, 2015.

[Ins] INSTITUT FÜR INFORMATIK: *Angebotene Dienste an den IfI Rechnerpools*. Besucht am 13.05.2016 `http://www.rz.ifi.lmu.de/Dienste/index.html`.

[ITW] ITWISSEN: *Hochverfügbarkeit*. Besucht am 24.07.2016 `http://www.itwissen.info/definition/lexikon/Hochverfuegbarkeit-high-avaiability-HA.html`.

[Jan] JANOTTA UND PARTNER: *IT-Asset Management*. Besucht am 20.09.2016 `http://www.n4dtn.de/it-asset-management.html`.

[Jv14] JIRSÍK, TOMÁŠ und PAVEL ČELEDA: *Identifying Operating System Using Flow-based Traffic Fingerprinting*. In: *Advances in Communication Networking*, Seiten 70–73. Springer, 2014.

[LAM77] LAMBOURNE, G. T. C.: *A Brief History of Fingerprints*. J. Forens. Sci. Soc, 17:95–98, 1977.

[Los07] LOSCHER, C.: *Strafverfolgung/Strafverfolgungsbehörden*, Februar 2007. Besucht am 15.05.2016 `http://www.lexexakt.de/glossar/strafverfolgung.php`.

[LR] LEIBNIZ-RECHENZENTRUM: *Das Münchner Wissenschaftsnetz*. Besucht am 14.05.2016 `https://www.lrz.de/services/netz/`.

[LR01] LEIBNIZ-RECHENZENTRUM: *LRZ Jahresbericht 2000*, April 2001. `https://www.lrz.de/wir/berichte/JB/JBer2000.pdf`.

[LR15] LEIBNIZ-RECHENZENTRUM: *LRZ Jahresbericht 2014*, Juni 2015. `https://www.lrz.de/wir/berichte/JB/JBer2014.pdf`.

[LR16] Leibniz-Rechenzentrum: *LRZ Präsentation*, 2016. LRZ-Interne Publikation.

[Mos10] Mossel, Siebren: *Passive OS detection by monitoring network flows.* 2010.

[Sch15] Schweizer, Markus: *Der direkte Nutzen von IT Asset Management*, Juli 2015. Besucht am 15.05.2016 `http://news.digicomp.ch/de/2015/07/06/der-direkte-nutzen-von-it-asset-management/`.

[Sta] Staatsanwaltschaft Niedersachsen: *Ablauf eines Ermittlungsverfahrens.* Besucht am 15.05.2016 `http://www.staatsanwaltschaften.niedersachsen.de/portal/live.php?navigation_id=22879&article_id=81138&_psmand=165`.

[TThCcC09] Tu, William, Priya Thangaraj, Jui hao Chiang und Tzi cker Chiueh: *Automated Service Discovery for Enterprise Network Management*, 2009.

[Ubu] Ubuntu Manpages: *PRADS - Passive Real-time Asset Detection System.* Besucht am 03.06.2016 `http://manpages.ubuntu.com/manpages/precise/man1/prads.1.html`.

[vMH13] von Eye, Felix, Stefan Metzger und Wolfgang Hommel: *Dr. Portscan: Ein Werkzeug für die automatisierte Portscan-Auswertung in komplexen Netzinfrastrukturen.* In: Paulsen, Christian (Herausgeber): *Sicherheit in vernetzten Systemen: 20. DFN Workshop*, Seiten C–1–C–21, Norderstedt, Deutschland, Januar 2013. Books on Demand.

[Wal14] Walonka, Christian: *FINGERPRINTING.* Algorithmen für Suche, Spiele und Geheimnisse, 2014.

[Wei11] Weisse, Günther K.: *Technische Kommunikationsüberwachung in Deutschland*, 2011. [Online; Stand 16. Mai 2016].

[Wö] Wöll, Peter E.: *Asset Management im Client/Server-Umfeld.* Besucht am 01.05.2016 `http://www.woell.ch/Asset_Management_im_Client_Server_Umfeld.pdf`.