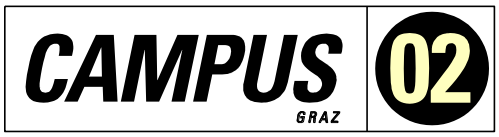
Masterarbeit

SICHERES REMOTE-MONITORING VERTEILTER PRODUKTIONSDATEN-SERVER

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von
Theresa Maria Ritzal, BSc
1510322004

betreut und begutachtet von
DI Peter Priller

Graz, im Jänner 2017 .....

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

Im Zuge der Bearbeitung meiner Masterarbeit haben mich einige Menschen besonders unterstützt, welche ich namentlich erwähnen möchte. Allen voran danke ich Herrn DI Peter Priller, welcher mir als Betreuer mit fachlichem Wissen und konstruktiven Vorschlägen immer weitergeholfen hat. Ein weiterer Dank gilt meinem Mann Niki, der mich besonders mental immer wieder aufgebaut hat. Ohne meine Eltern, besonders meine Mama Michaela Koch, hätte ich den Fertigstellungstermin meiner Arbeit nicht einhalten können – vielen Dank, dass ihr in dieser Zeit so liebevoll auf Paul aufgepasst habt. Ein großes Dankeschön auch an meine Schwester Anna-Sophie Koch, die meine Arbeit auf Rechtschreibfehler überprüft hat. Zum Schluss möchte ich noch meinen besonderen Dank an die Firma LOGICDATA, vor allem Herrn DI (FH) Walter Weinberger, MBA, aussprechen, die mir viel Zeit, fachliche Beratung und das Testsystem zur Verfügung gestellt haben.

KURZFASSUNG

Die Sicherstellung dauerhafter Verfügbarkeit von produktionsnahen Servern ist für betroffene Firmen eine zunehmende Herausforderung. Bei einem Ausfall, sei er auch nicht von langer Dauer, kommt es bereits zu erheblichen zeitlichen Verzögerungen. Schwerwiegender sind allerdings die monetären Verluste, die durch unzureichende Überwachung der Server entstehen können.

Aus diesem Grund liegt ein großes Augenmerk dieser Arbeit auf der Evaluierung und Einführung einer geeigneten Remote-Monitoring-Software für Produktionsdaten-Server, welche sich bei Lohnfertigern der Firma LOGICDATA befinden. Dabei wurden verschiedene Lösungen in einer Testumgebung untersucht und miteinander verglichen. Die zuvor definierten Kriterien wurden von zwei Software-Lösungen weitestgehend erfüllt, welche in der Arbeit detaillierter beschrieben werden.

Da die betroffenen Server örtlich verteilt sind, muss auch eine sichere Übertragung der Überwachungsdaten gewährleistet sein. Infolgedessen wurde zusätzlich die bestehende VPN-Verbindung zu den Lohnfertiger-Servern untersucht und mit anderen Lösungen bzw. Protokollen verglichen. Eine mögliche Änderungsempfehlung wurde im Zuge dieser Masterarbeit entworfen, die allerdings gezeigt hat, dass die derzeit eingesetzte Lösung in diesem Anwendungsbereich durchaus Angriffe von außen verhindern kann.

Nach dem Evaluierungsprozess in der Testumgebung wurden die ausgewählten Monitoring-Tools auf den tatsächlich eingesetzten Produktionsdaten-Servern erprobt. Die Überwachungsmöglichkeiten werden in Zukunft das Risiko eines Ausfalls bzw. auch die finanziellen Auswirkungen eines solchen für die Firma LOGICDATA minimieren.

ABSTRACT

To ensure the permanent availability of production data server systems, companies like LOGICDATA are obligated to take action in advance to prevent any data failure or loss. Moreover a short interruption of data availability can cause delays in production schedules. Even more serious is the monetary impact for the company, which can result from insufficient server monitoring.

For this reason one focus of this thesis lies in the evaluation of appropriate software solutions for the remote monitoring of LOGICDATA's data servers at the manufacturing sites. As a result, recommendations for the implementation of the two selected monitoring tools for the real environment are given. Due to the fact that LOGICDATA's servers are scattered around the globe, a secure transfer of the exchanged data must be guaranteed under all circumstances. Therefor the second focus is put on the examination and comparison of the existing VPN solution to other state-of-the-art protocols.

On the one hand, the implementation of the selected monitoring software will enable LOGICDATA to minimise the risk of data failure or loss and monetary impacts. On the other hand, the analysis of the currently used VPN solution showed that it is protective against attacks in this field of application.

INHALTSVERZEICHNIS

1 EINLEITUNG

1.1 Problemstellung

Durch die steigenden Zuverlässigkeitsanforderungen an Server, welche für einen reibungslosen Produktionsprozess mitverantwortlich sind, ist es notwendig, diese rund um die Uhr zu überwachen. Bei Ausfall eines Servers kann es je nach Einsatzart zu Stillständen kommen oder Störungen verursachen, welche hohe Kosten mit sich bringen. Dieser Zustand kann mit einem geeigneten Monitoring-System verhindert werden, welches Parameter wie Netzwerkqualität der VPN-Verbindung, Speicherauslastung und andere prüfen und überwachen soll.

Nicht nur eine gut funktionierende Software ist jedoch für Remote-Monitoring von Servern von Nöten, sondern besonders die Datenverbindung zwischen der Monitoring-Software auf der einen Seite und dem Server auf der anderen Seite. Dabei muss der Fokus vor allem auf die Aspekte Sicherheit und Stabilität gelegt werden.

Im Fall der Firma LOGICDATA geht es um Server, welche sich nicht vor Ort befinden, sondern über VPN (Virtuelles Privates Netzwerk) mit dem Standort in Deutschlandsberg verbunden sind. Auf diesen Servern befinden sich produktionsrelevante Daten (wie z.B. Auftragsdaten und Testdaten), was daher bei einem Ausfall des Servers zu einem Stillstand der Produktion führen würde. Dieser Umstand veranlasste die Firma LOGICDATA dazu, das Monitoring-Konzept zu optimieren (siehe Kapitel 1.3.3 Aktuelles Monitoring (Hydra)).

1.2 Wissenschaftliche Fragestellung & Zielsetzung

Im Theorieteil dieser Arbeit wird besonders auf VPN-Verbindungen, deren Sicherheit und Alternativen eingegangen. Folgende Fragen sollen diesbezüglich beantwortet werden:

- Welche Voraussetzungen gibt es für den Aufbau einer VPN-Verbindung? Welche Topologien und Typen sind bewährt und wo werden diese eingesetzt?
- Welche Vor- bzw. Nachteile entstehen durch unterschiedliche Verschlüsselungsarten? Wo ist ihr Einsatz sinnvoll?
- Welche Angriffe auf VPN-Verbindungen sind bekannt und wie können diese verhindert werden?
- Wie wird die Sicherheit von VPN-Verbindungen in Zukunft gewährleistet? Was ist State-of-the-Art? Welche Alternativen gibt es für bewährte VPN-Lösungen?
- Ist eine Technologieänderung der bereits bestehenden VPN-Lösung der Firma LOGICDATA notwendig? Welche Alternativen kommen dafür in Frage und welche Auswirkung hätten diese auf die Verschlüsselung? Welche Vor- bzw. Nachteile entstehen durch eine Umstellung?

Im praktischen Abschnitt liegt das Augenmerk auf dem Remote-Monitoring von Servern der Firma LOGICDATA. Eine Evaluierung und Einführung eines geeigneten Systems bildet hier den Mittelpunkt. Durch Testaufbauten soll geprüft werden, mit welcher Methode bzw. mit welchem Werkzeug sich die

oben genannten Parameter am besten prüfen bzw. messen lassen. Dabei sind folgende Fragestellungen relevant:

- Mit welcher Technologie kann ein Live-Monitoring realisiert werden?
- Welche Software ist für die Anforderungen geeignet?

1.3 Ausgangssituation – Server Monitoring

Wie in Kapitel 1.1 Problemstellung erwähnt, steigt die Anforderung an die Zuverlässigkeit von Servern , die dauerhaft garantiert werden muss. In den nächsten Kapiteln wird nun die Firma LOGICDATA im Detail vorgestellt, die wirtschaftlichen Aspekte des Monitorings hervorgehoben und das derzeitige Monitoring-System und die damit auftretenden Schwierigkeiten beschrieben.

1.3.1 Vorstellung der Firma LOGICDATA

LOGICDATA ist ein international tätiges Technologieunternehmen, das sich mit der Entwicklung und Herstellung von mikroprozessorbasierten Steuerungen und Bedienelementen für die Möbelindustrie beschäftigt. Am Standort Deutschlandsberg werden derzeit rund 110 Mitarbeiter/Innen beschäftigt, wovon ca. 60 Prozent in der Forschung und Entwicklung tätig sind. Aufgrund seiner Innovationsstärke konnte sich LOGICDATA in den vergangenen Jahren nachhaltig und erfolgreich am Markt für höhenverstellbare Möbel etablieren.

LOGICDATA wurde 1994 von DI Walter Koch als Einzelunternehmen gegründet und 1997 in eine GmbH umgewandelt. Seit 2002 sind mit DI Roland Koo und Mag. Judith Koo zwei Finanzinvestoren an der LOGICDATA Electronic & Software Entwicklungs GmbH beteiligt. Die Eigentümerverhältnisse lassen sich wie folgt abbilden:

- DI Walter Koch: 80 Prozent
- DI Roland Koo: 10,25 Prozent
- Mag. Judith Koo: 9,75 Prozent

Die Firma LOGICDATA ist zu 100 Prozent Anteilseigner der LDI Electronic-Vertriebs GmbH (Sitz in Deutschlandsberg) und der LOGICDATA d.o.o. (Sitz in Maribor, Slowenien). Die LDI Electronic-Vertriebs GmbH ihrerseits ist wiederum zu 100 Prozent Anteilseigner der LOGICDATA North America Inc., deren Unternehmenssitz sich in Grand Rapids (Michigan, USA) befindet.

Die Geschäftsführung von LOGICDATA liegt bei DI Walter Koch, DI Johannes Gradwohl, MBA, DI Stefan Lukas, MBA und Dr. Jörg Schweiger, MSc. Ihnen beratend zur Seite steht ein freiwillig gebildeter Beirat, dem auch Roland und Judith Koo wie auch andere Führungspersönlichkeiten der steirischen Wirtschaft angehören. Mit der Einführung des Managementteams im Jahr 2006 wurde die Führungskompetenz bei LOGICDATA weiter ausgeprägt und dadurch ein Mehrwert an strategischer Kompetenz im Unternehmen generiert.

Die von LOGICDATA angebotenen Produkte sind darauf ausgerichtet, ergonomisch hochwertige Arbeits- und Lebensbedingungen zu ermöglichen. Um dieses Ziel zu erreichen sind im Unternehmen folgende zwei Geschäftsbereiche verankert:

In der Büromöbelbranche bietet „LogicOffice“ Steuerungen und Handschalter für elektronisch höhenverstellbare Sitz-Steh-Arbeitsplätze an. „LogicHome“ konzentriert sich auf Steuerungstechnologie und Bedienelemente für Komfortbetten und Relaxsessel. In beiden Sparten werden den Kunden/Innen größtenteils individuell maßgeschneiderte Lösungen zur Verfügung gestellt. Diese Flexibilität und Professionalität sind wichtige Kernkompetenzen von LOGICDATA.

LOGICDATA konnte in den letzten Jahren durch seine Innovations- und Verkaufsstärke den größten und wichtigsten Hersteller für Komfortbetten am US-amerikanischen Bettenmarkt gewinnen und damit den Umsatz des Wirtschaftsjahres 2012/13 von 26 Mio. EUR auf 42 Mio. EUR für das Wirtschaftsjahr 2013/14 steigern.

Abbildung 1: Antrieb LOGICdrive, Quelle: LOGICDATA (2016), Onlinequelle [16.05.2016].

1.3.2 Wirtschaftliche Aspekte des Monitorings

Als Motivation zur Einführung eines Server-Monitorings gelten vor allem das Service Level Agreement (SLA) sowie die Vermeidung eines Ausfalls der Server, da ein solcher Schaden auch andere Tätigkeitsbereiche beeinflussen würde. Diese beiden Themen werden auf den kommenden Seiten näher erläutert.

1.3.2.1 Service Level Agreement

Unter Service Level Agreement wird eine Vereinbarung zwischen Dienstleistern und Kunden verstanden, die sich auf die Qualität des Service bezieht (siehe Abbildung 2).[1] Konkret am Beispiel der Firma LOGICDATA ist der Dienstleister die IT-Abteilung, der Kunde die Abteilung Product-Engineering und die Dienstleistung die Betreuung und Wartung der Lohnfertiger-Server. Es besteht sozusagen ein internes SLA. Dies ist notwendig, um Leistungsparameter und Verantwortlichkeiten von Seiten der IT (wie z.B. Reaktionszeiten oder Verfügbarkeiten) festzulegen und abzugrenzen. Am Ende zählt dabei, dass die Wünsche und Interessen des Endkunden bestmöglich erfüllt werden.

[1] Vgl. Schrey (2006), S.29.

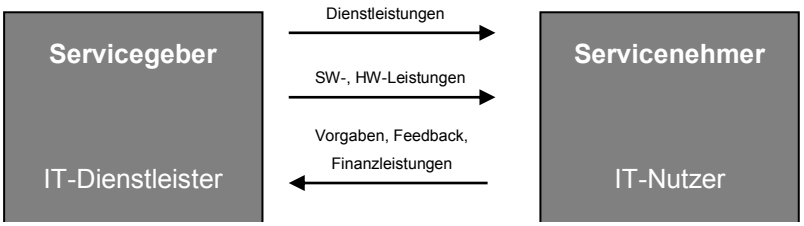


Abbildung 2: SLA als Schnittstelle zwischen Servicegeber und -nehmer, angelehnt an: Mann (2004), S.52.

Am Beginn dieser Vereinbarung erfolgt die Festlegung des Servicegrades. Dabei gibt der Kunde das Minimum an Service vor. Nach oben hin lässt sich der Servicegrad beliebig erhöhen. Allerdings ist eine zu hohe Serviceleistung nicht mehr rentabel. Als Dienstleister sollte man außerdem beachten, welche Leistungen auch bei hoher Auslastung noch durchführbar sind. Die vereinbarte Serviceleistung sollte jederzeit und mit gleichbleibender Qualität durchgeführt werden können.[2]

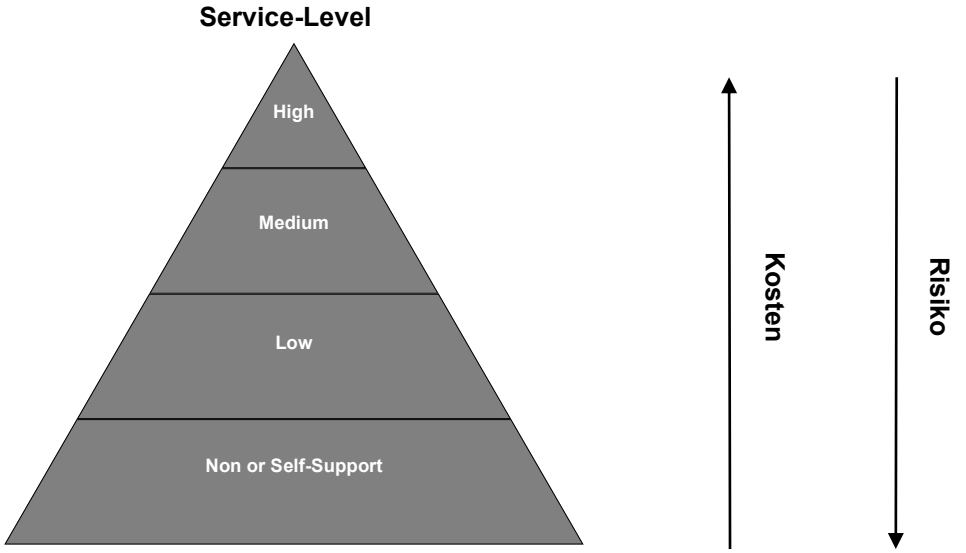


Abbildung 3: Service-Level-Stufen, angelehnt an: Hiles (2002), S.7.

Grundsätzlich lassen sich drei Arten von SLAs unterscheiden:

1. Inputorientierte SLAs („Inputstandardisierung“): die Vereinheitlichung von Inputfaktoren (z.B. Qualifikation der Mitarbeiter) innerhalb des Dienstleistungsprozesses
2. Verrichtungs- und prozessorientierte SLAs („Prozessstandardisierung“): legen Kennzahlen fest, die zur Beurteilung des Leistungserstellungsprozesses dienen (z.B. Verfügbarkeit in bestimmten Zeitfenstern).
3. Outputorientierte SLAs („Outputstandardisierung“): legen Kennzahlen fest, die zur Beurteilung der tatsächlich erbrachten Dienstleistung dienen (z.B. Verfügbarkeitsgarantie von 99%).

Diese beschriebenen Arten lassen sich für eine optimale Qualität je nach Einsatzfall auch kombinieren.[3]

[2] Vgl. Hiles (2002), S.5ff.

[3] Vgl. Schrey (2006), S.30f.

Ein besonderer Vorteil der SLAs ist aus Sicht der Kunden besonders die Senkung von Überwachungskosten, da sie sich darauf verlassen können, dass die vereinbarten Serviceleistungen auch durchgeführt werden.[4] Trotzdem muss auch auf die Probleme von SLAs hingewiesen werden.

1. Die Qualität und Erstellungskosten von Dienstleistungen können im Gegensatz zu Sachleistungen schwer ermittelt werden.
2. Wichtige Aspekte der Servicequalität sind nicht oder schwer quantifizierbar (z.B. Freundlichkeit des Personals).
3. Zur Beurteilung der Dienstleistungsqualität zählen viele subjektive Parameter, die sehr von der persönlichen Stimmung und Erwartung des Kunden abhängen.

Zusammenfassend ist anzumerken, dass SLAs im Dienstleistungsbereich notwendig für eine gute und langfristige Zusammenarbeit sind.[5]

1.3.2.2 Auswirkungen eines Lohnfertiger-Server-Ausfalls

Wie bereits im vorangehenden Kapitel beschrieben, muss eine präzise Vereinbarung über die Wartung der Server getroffen werden, um Ausfälle und andere Komplikationen möglichst zu vermeiden. Diese können mit einer geeigneten Monitoring-Software erkannt werden.

Konkret werden bei der Firma LOGICDATA auf jedem Lohnfertiger-Server Aufträge und Testdaten gespeichert, sowie die Steuerung und Überwachung von Testgeräten durchgeführt. Dabei kann es nun zu folgenden Fehlerfällen kommen:

Bei einem **Totalausfall**, d.h. der Server lässt sich durch Fernwartung nicht mehr reparieren, muss sich Personal an den Ort des jeweiligen Lohnfertigers begeben, um diesen entweder auszutauschen oder wiederherzustellen. Dies verursacht neben Reisekosten von ca. €1000,-- (je nach Aufwand und Personaleinsatz) vor allem auch Kosten für LOGICDATA (im Schnitt ca. €300.000,-- bis €400.000,-- pro Woche beim größten Lohnfertiger), da dadurch Lücken in der Produktion entstehen. Dies ist aber nur dann der Fall, wenn die Fertigung inzwischen nicht mit anderen Aufträgen beschäftigt werden kann.

Kommt es zu einem **kurzzeitigen Ausfall** (z.B. ein Neustart des Servers durch nicht gewolltes Installieren von Updates), ist auch hier mit einer Verzögerung in der Produktion zu rechnen, insbesondere dann, wenn der jeweilige Lohnfertiger 24/7 verplant ist. Allerdings lässt sich die Art von Ausfällen gut und schnell durch Fernwartung beheben. Eine Einrichtung einer IT-Hotline bei LOGICDATA, welche rund um die Uhr erreichbar ist, führt zu einer hohen Reaktionsgeschwindigkeit.

Unabhängig von der Art des Ausfalls kommt es bei einer Unterbrechung des Testzyklus immer zu Problemen, da die Ergebnisse von bereits vorher durchgeführten Tests auf der Steuerung nicht mehr gespeichert wurden und somit die Fertigung nicht verlassen können. In diesem Fall muss das Testen aller betroffenen Steuerung ein weiteres Mal wiederholt werden.

---

[4] Vgl. Schrey (2006), S.36.

[5] Vgl. Schrey (2006), S.39f.

Generell wurde im Hinblick auf unvorhersehbare Ereignisse für die gesamte Herstellungsdauer (inkludiert das Versenden der Teile an die Lohnfertiger, die Produktions- und Testzeit, sowie die Rücksendung an LOGICDATA) eine Pufferzeit von ein bis zwei Wochen eingeplant, um den Kunden möglichst zufriedenzustellen. Sollte trotz aller Maßnahmen ein längerer Ausfall auftreten, kommt es zur Verzögerung von Aufträgen, da generell bis an die Kapazitätsgrenze geplant wird.

1.3.3 Aktuelles Monitoring (Hydra)

Um die heterogene Netzwerk-Landschaft der Firma LOGICDATA bestmöglich überwachen zu können, wurde 2009 die Software „Hydra“ intern entwickelt, welche auf einem Server am Standort Deutschlandsberg installiert ist und von dort aus die Server bei Lohnfertigern überwacht. Der Zweck dieses Tools ist es, die Verfügbarkeit sämtlicher Netzwerkkomponenten zu erhöhen und bei kritischen Vorfällen Verantwortliche via Email zu benachrichtigen, um eine hohe Reaktionsgeschwindigkeit zu ermöglichen. Für die Entwicklung der Software wurde .NET gewählt, im Speziellen die Programmiersprache C#.

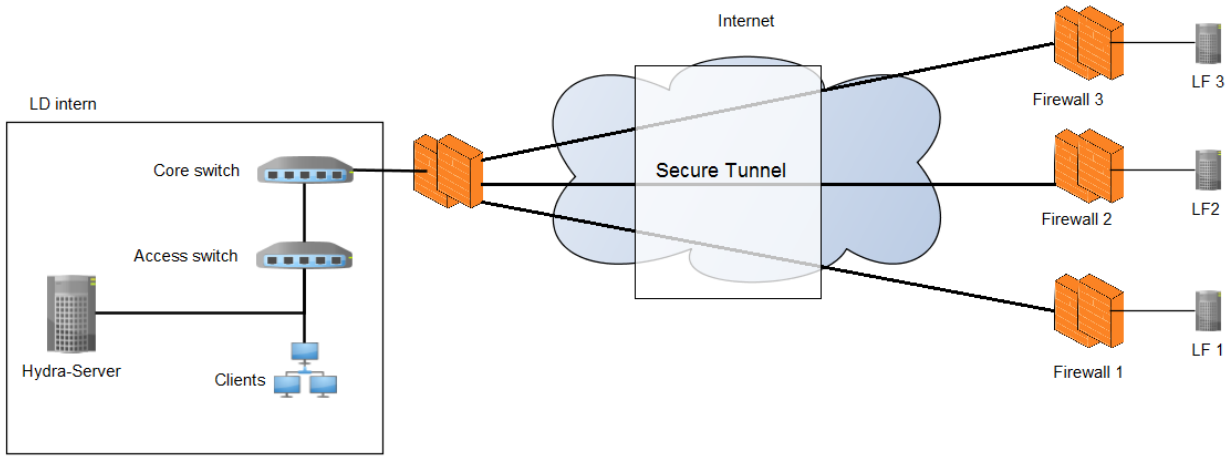


Abbildung 4: Netzwerkdiagramm zu Lohnfertigern, Quelle: Eigene Darstellung.

1.3.3.1 Aufbau und Funktionen

Hydra besteht aus mehreren Klassen, wobei jeweils eine Klasse für eine Aufgabe verantwortlich ist. Dabei übernimmt die „Control“-Klasse den Hauptteil der Software. Sie ist u.a. für das Starten und Stoppen der Event-Handler und für die E-Mail-Mitteilungen zuständig.

Für die Abfragen auf den Servern wurden SNMP (Simple Network Management Protocol) und WMI (Windows Management Instrumentation) zur Hilfe genommen. Dabei werden folgende Ausformungen unterschieden:

WMI:

- WMI Query: Die Software fragt regelmäßig einen Server ab und erhält eine Antwort
- WMI Event: Die Software registriert einen Grenzwert auf einem Server. In diesem Fall schickt der Server von sich aus die Daten zur Hydra

SNMP:

- SNMP Query: analog zur WMI Query
- SNMP Traps: Der „SNMP Trap Listener“ ist ein einzelner eigener Thread, welcher auf dem Port 162 auf alle einkommenden Traps reagiert. Die Traps werden auf dem jeweiligen Server registriert (nicht von der Hydra, sondern direkt am Server) und von diesem aus zur Hydra geschickt.

Eine weitere Komponente ist die Datenbank. Hierfür wurde ein Microsoft SQL Server 2005 verwendet, welcher in Deutschlandsberg lokalisiert ist. In der Datenbank werden alle Events (unabhängig von der Gewichtung) gespeichert. Ebenso wurde ein eigens angelegter Datenbank-Benutzer eingerichtet, welcher sich mit der Datenbank verbindet. Somit muss auf dem Gerät, auf dem die Software ausgeführt wird, kein Benutzer angemeldet sein. Dies ermöglicht die Ausführung der Software als Service.

Für die Abfragen an den Geräten wurden, wie oben beschrieben, WMI und SNMP verwendet. Dabei ist es wichtig zu erwähnen, dass diverse Netzwerk-Komponenten (z.B. Firewalls, Switches, etc.) nur bis SNMP v2c unterstützt werden. Aufgrund dessen mussten folgende Maßnahmen getroffen werden, um trotzdem sicher zu sein:

- SNMP wird ausschließlich im internen Netzwerk genutzt. Die Ports (161 und 162) für ankommende SNMP-Abfragen wurden auf der Firewall blockiert.
- Bei den „Community Strings“ wird nicht der Standardname „public“ verwendet.
- Die zu überprüfenden Netzwerkkomponenten wurden so konfiguriert, dass nur Anfragen des Monitoring-Hosts akzeptiert werden.
- Externe SNMP-Vorgänge werden nur über einen IPSec (Internet Protocol Security) VPN-Tunnel durchgeführt.

Essentiell ist bei der Software die Benachrichtigungsfunktion, um schnellstmöglich die Verantwortlichen zu informieren. Dafür wurden für alle Abfragen an den Geräten Grenzwerte definiert. Bei Abweichung dieser wird eine Benachrichtigung mit einer Beschreibung sowie den abgefragten Werten verschickt.

1.3.3.2 Konfiguration und GUI

Für die Konfiguration müssen die zu überwachenden Geräte mittels XML-Konfigurationsdateien definiert werden. Für jede Komponente müssen folgende Werte angegeben werden:

- **Host**: Gibt die Adresse des zu überwachenden Gerätes an. Dies kann eine IP Adresse oder ein Hostname sein.
- **Computername**: Wird verwendet um das Gerät für die Überwachung zu identifizieren (da es sein kann, dass sich die Adresse irgendwann ändert).
- **Interval**: Gibt das Polling-Intervall in Millisekunden an.
- **Username**: Gibt den Benutzer an, mit dem die Verbindung durchgeführt werden soll (nur für Windows Maschinen).
- **Passwort**: Gibt das Passwort für den verwendeten Benutzer an (nur für Windows Maschinen).

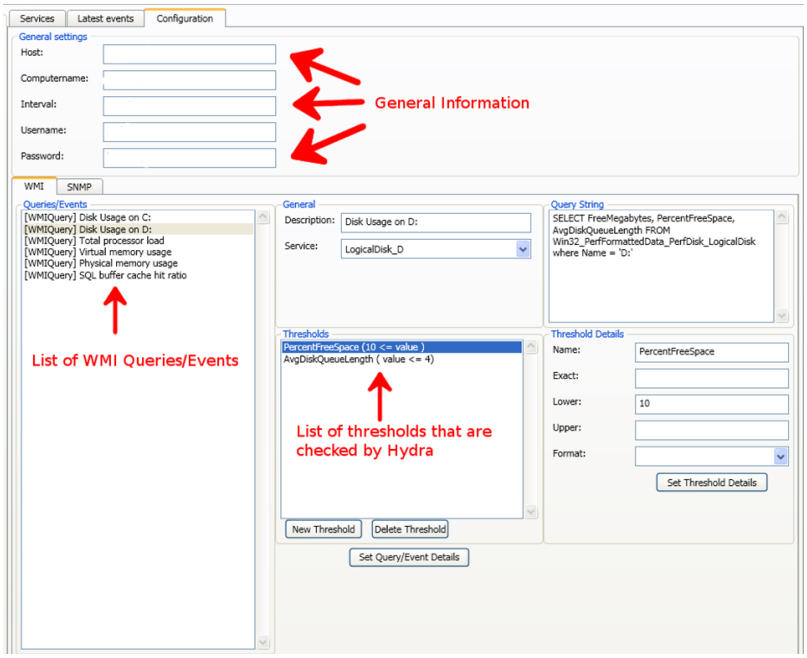


Abbildung 5: GUI Hydra, Quelle: Eigene Darstellung.

Aufgrund der wachsenden Zahl an zu überwachenden Netzwerkkomponenten ist eine beliebige Erweiterung in der Hydra sehr umständlich. Dies veranlasste die Firma LOGICDATA zum Wechsel auf eine andere Monitoring-Software. Neben der Erweiterung der technischen Möglichkeiten steht auch eine gute Benutzerfreundlichkeit im Vordergrund bei der Auswahl der neuen Software.

2 GRUNDLAGEN

International agierende Unternehmen stehen vor der immer größer werdenden Herausforderung, alle Mitarbeiter möglichst gut miteinander zu vernetzen. Dabei reichen Hilfsmittel wie Telefone und Faxgeräte schon lange nicht mehr aus. Alle Möglichkeiten, die Kommunikation untereinander zu verbessern, müssen ausgeschöpft werden. Dabei ist der Einsatz eines „Virtuellen Privaten Netzwerkes“ (VPN) sehr sinnvoll, welches einzelne Standorte über das Internet verbindet (siehe Abbildung 6).[6] Entscheidende Faktoren für die weltweit vielverbreitete Technologie ist vor allem dem Ausbau der Internetverbindungen und den sinkenden Kosten dieser Verbindungen zu verdanken.[7]

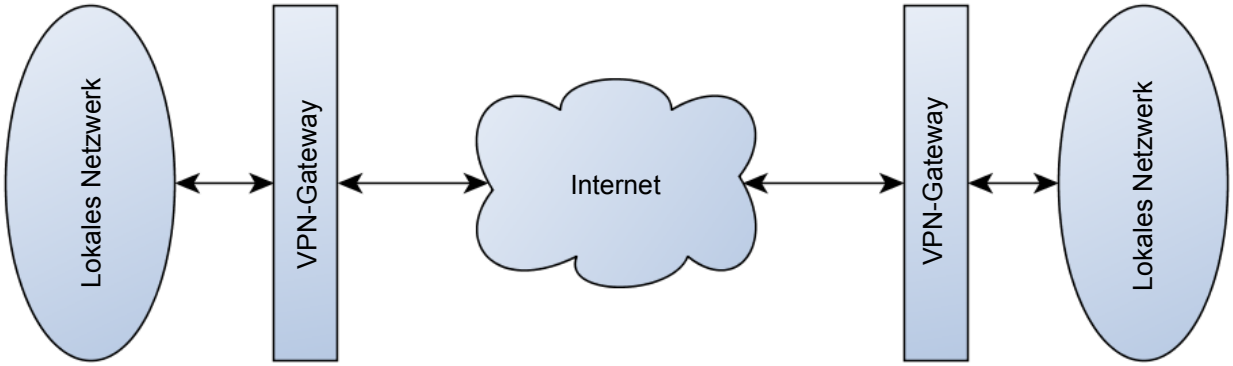


Abbildung 6: Funktionsprinzip VPN, Quelle: In Anlehnung an: Aebi (2004), S.77.

2.1 Nutzen von VPN

Durch geografische Gegebenheiten mussten früher weite Strecken (z.B. Verbindung zu einer Außenstelle) mit Standleitungen überbrückt werden. Diese waren zwar sicher, aber für weite Distanzen sehr teuer. Die Nutzung des Internets reduziert nicht nur die Kosten, sondern bringt auch noch andere Vorteile mit sich.[8]

2.1.1 Flexibilität

Wie bereits erwähnt, wird der Nutzen einer VPN-Verbindung besonders am Beispiel geografisch verteilter Standorte ersichtlich. Ist erst einmal eine Verbindung eingerichtet, können beliebig viele Benutzer auf Daten des zentralen Standortes zugreifen. Hier bedarf es keiner weiteren Infrastruktur und eine schnelle Einrichtung der VPN-Verbindung am jeweiligen Gerät kann zeitnah durchgeführt werden. Ebenso das gegenteilige Szenario – die Verbindungen können auch einfach wieder zurückgesetzt bzw. getrennt werden. Diese Flexibilität kann besonders bei großen Organisationen eine große Erleichterung darstellen.[9]

[6] Vgl. Reiser (2000), S.83.

[7] Vgl. Tiller (2002), S.216.

[8] Vgl. Aebi (2004), S.76f.

[9] Vgl. Tiller (2002), S.277ff.

2.1.2 Kostenaspekte

Nicht nur für die Kommunikation zwischen Mitarbeitern, die sich nicht am zentralen Standort befinden, werden VPN-Verbindungen eingerichtet. Manchmal ist es auch nötig, eine sichere, temporäre Verbindung für einen Datenaustausch aufzubauen. Müsste man dafür zusätzliche Netzwerkkomponenten (z.B. Router, Switches, etc.) zukaufen, würde das eine große Investition bedeuten. Diese Kosten können durch die einfache Konfiguration von VPN-Verbindungen vermieden werden.[10]

2.1.3 Sicherheit

Bevor auf den Aspekt der Sicherheit in Bezug auf VPN-Verbindungen eingegangen wird, werden zuvor noch allgemeine Begriffe zum Thema „Sicherheit" näher erläutert.

Obwohl sich Sicherheit nur schwer quantitativ feststellen lässt, gibt es doch qualitative Eigenschaften, die die Beurteilung von möglichen Gefahren erleichtert.

Begriff	Erläuterung
Vertraulichkeit	Daten sind für Dritte nicht erreichbar. „*Unerlaubte Zugriffe sind nicht möglich*"[11]
Integrität	Darunter werden sowohl korrekte Daten, als auch die korrekte Funktion von Hardware- und Softwarekomponenten verstanden. In Bezug auf Daten ist darauf hinzuweisen, dass Integrität nur dann gegeben ist, wenn keine unerwünschten Veränderungen vorgenommen wurden.
Verfügbarkeit	Unter Verfügbarkeit wird die tatsächlich zur Verfügung stehende Zeit der Netzwerkkomponenten verstanden. Diese wird als Verhältnis zwischen der Zeit angegeben, in der die Komponenten ordnungsgemäß funktionieren und der Gesamtzeit (Ausfallszeit + verfügbare Zeit). Die Verfügbarkeit wird üblicherweise in Prozent angegeben.
Authentifikation	Durch Authentifikation kann sichergestellt werden, dass nur berechtigte Personen Zugriff zu einem System oder Daten bekommen. Dies kann auf unterschiedliche Arten erfolgen[12]: • *„Durch Wissen (Passwort, PIN, …),* • *durch Besitz (Schlüssel, Scheckkarte, …),* • *durch Merkmale (Fingerabdruck, Iris-Erkennung, …)."* Bei Kombinierung dieser Arten wird von einer „starken Authentisierung" gesprochen.
Verbindlichkeit	Bei rechtsverbindlichen Geschäften, welche über Netzwerkkomponenten abgewickelt

[10] Vgl. Tiller (2002), S.275f.

[11] Aebi (2004), S.12.

[12] Aebi (2004), S.13.

	werden (z.B. elektronischer Handel), muss eine inhaltliche Zurechenbarkeit gewährleistet werden.
Zugriffskontrolle	Verhinderung von unberechtigten Zugriffen auf Programme oder Daten. Diese Mechanismen finden nach der Authentisierung statt.
Anonymität	Muss in diversen Fällen aus datenschutzrechtlichen Gründen gewährleistet sein, um persönliche Daten der Akteure zu schützen.

Tabelle 1: Sicherheitsbegriffe, vgl. Aebi (2004), S.12ff.

Der Begriff der Sicherheit ist in der Praxis nicht immer positiv behaftet. Durch zusätzliche Kosten und manchmal einschränkende Benutzerfreundlichkeit ist es oft eine Herausforderung für Verantwortliche, das richtige Maß an Sicherheitsmaßnahmen zu finden. Oft werden diese Maßnahmen erst nach entstandenem Schaden eingeführt.[13]

Dabei sollten folgende drei Aspekte zum Thema IT-Sicherheit beachtet werden[14]:

- Wirtschaftliche Aspekte
    - Hierbei gilt es, einen Kompromiss (siehe Abbildung 7) zwischen den Kosten zur Sicherheitserhöhung und den Kosten, die durch einen Schadensfall entstehen, zu finden.

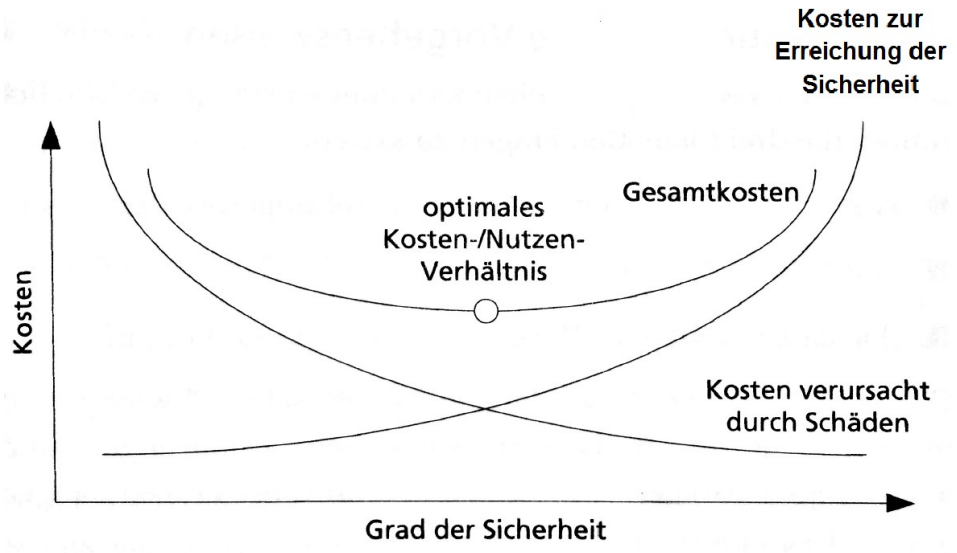


Abbildung 7: Optimales Kosten-/Nutzen-Verhältnis, Quelle: Aebi (2004), S.15.

- Juristische Aspekte
    - Oft sind Sicherheitsmaßnahmen durch Gesetze (z.B. Datenschutzgesetze) vorgegeben und werden dem Unternehmen somit „aufgezwungen“. Auch diese müssen bei Betrachtung der IT-Sicherheit berücksichtigt werden.
- Psychologische Aspekte

[13] Vgl. Aebi (2004), S.14f.

[14] Vgl. Aebi (2004), S.15f.

- Investitionen in die Sicherheit der IT müssen unter unsicheren Annahmen entschieden werden. Dabei muss bei den Verantwortlichen ein entsprechendes Problembewusstsein vorhanden sein, um sich überhaupt für Maßnahmen in diesem Bereich entscheiden zu können.

Obwohl der Einsatz von VPN-Verbindungen eine Erleichterung u.a. in den Bereichen Flexibilität und Kosten gebracht hat, ist auch das Thema der Sicherheit unbedingt zu beachten. Die Verwendung unterschiedlicher VPN-Technologien (z.B. IPSec, SSL, PPTP, etc.) stellt unterschiedliche Stufen von Sicherheit dar.[15] Wie unterschiedlich Angriffe auf VPN-Verbindungen durchgeführt werden, wird im Kapitel 4.1 „Mögliche Angriffsszenarien auf VPN-Verbindungen“ näher erläutert.

Im Allgemeinen wird die Sicherheit von VPN durch die Kombination aus Verschlüsselung, Tunneling[16] und Authentifizierung sichergestellt.[17] Zu einer Risikoerhöhung kann es bei Fehlern im Design, der Konfiguration oder der Instandhaltung der VPN-Verbindung kommen. Auch wenn auf diese Aspekte besonderes Augenmerk gelegt wird, kann ein Angriff von außen trotzdem nicht vollständig verhindert werden. Das Risiko eines solchen Schadenfalls kann allerdings durch Maßnahmen, wie die Minimierung der zur Verfügung gestellten Daten oder die Verschlüsselung von Daten und Laufwerken, noch weiter reduziert werden. Dies geht jedoch oft mit der Einschränkung der Benutzerfreundlichkeit einher.[18]

2.2 VPN-Typen

Verschiedene Typen von VPN-Verbindungen werden für unterschiedliche Einsatzzwecke unterschieden. Diese werden auf den folgenden Seiten näher erläutert.

2.2.1 Remote-Access VPN

Ein Remote-Access VPN besteht einerseits aus dem Server, und andererseits aus dem Client, welcher wiederum aus einer Client-Software (z.B. Textverarbeitungsprogramme, CAD-Programme oder diverse andere Programme, welche auf die Daten über die VPN-Verbindung zugreifen) und einer mobilen Client-Hardware (z.B. Laptop, Smartphone, etc.) besteht (siehe Abbildung 8). Diese Form des VPNs wird besonders häufig bei Mitarbeitern verwendet, welche auf Dienstreisen immer wieder auf firmeninterne Daten zugreifen müssen („mobile Klienten“).[19] Diese Clients können nicht anhand ihrer IP-Adresse identifiziert werden, da sich diese durch unterschiedliche Internetverbindungen ständig ändert. Hier erfolgt die Authentifizierung mittels Benutzername und Passwort.[20]

[15] Vgl. Tiller (2002), S.215.

[16] Tunneling…verschlüsselte Verbindung zwischen zwei Endpunkten.

[17] Vgl. Pasley (2002), S.173.

[18] Vgl. Tiller (2002), S.237.

[19] Vgl. Pasley (2002), S.151ff.

[20] Vgl. Hekerens (2001), Online-Quelle [1.Juli.2016].

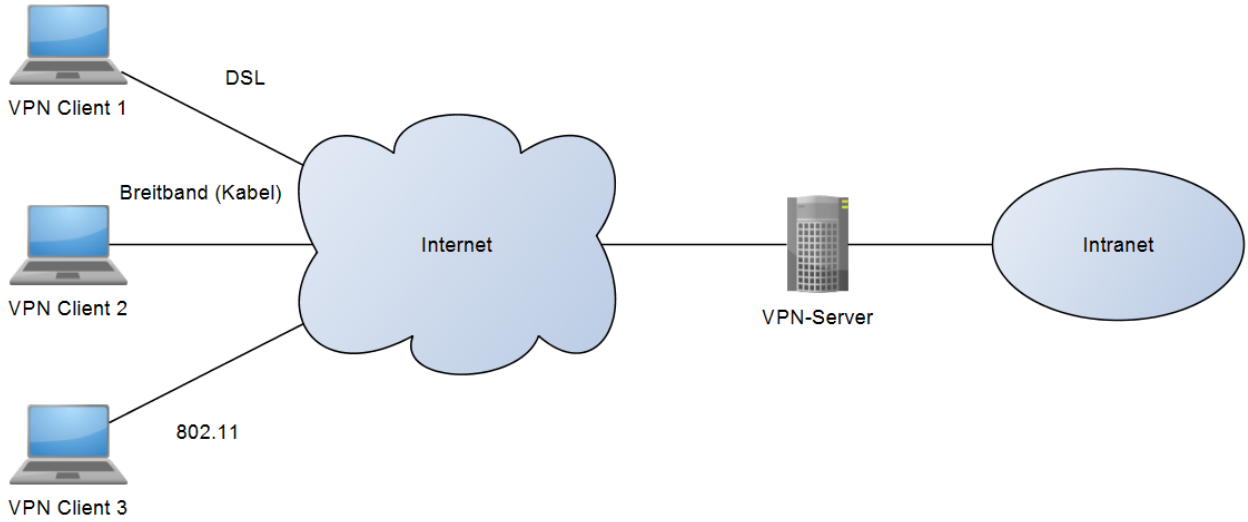


Abbildung 8: Remote Access VPN, Quelle: In Anlehnung an: Lipp (2001), S.42.

Für den Einsatz als VPN-Client und –Server sollten u.a. folgende Anforderungen erfüllt sein:

Client	Daten-/Laufwerk-Verschlüsselung
	Hohe Leistungsfähigkeit des Clients muss auch bei Verschlüsselung der Daten gegeben sein
	Möglichkeit, die VPN-Konfiguration für Benutzer zu sperren
	Einhaltung von aktuellen VPN-Standards
Server	Skalierbarkeit
	Hohe Verfügbarkeit
	Einhaltung von aktuellen VPN-Standards

Tabelle 2: Anforderungen an VPN-Client und –Server, vgl. Pasley (2002), S.157f.

2.2.2 Branch-Office VPN

Diese Art der VPN-Verbindung (auch „Intranet VPN“ genannt) ist, wie der Name schon andeutet, für die Anbindung von Außenstellen an eine zentrale Stelle gedacht, von der aus Benutzer aus der Ferne auf Daten und diverse Server in der Zentrale mühelos zugreifen bzw. diese nutzen können. Die Rechner sind über VPN so verbunden, als wären sie direkt in der Zentrale selbst positioniert (siehe Abbildung 9).[21] Dabei sollten die Kosten für zusätzliche Netzwerk-Infrastruktur möglichst minimiert und die Datenübertragungsmenge maximiert werden. Die technischen Vorteile dieses Typs sind vor allem der Einsatz flexibler Topologien (siehe Kapitel 2.3 VPN Topologien) und das einfach Hinzufügen von neuen Außenstellen.[22]

[21] Vgl. Hekerens (2001), Online-Quelle [1.Juli.2016].

[22] Vgl. Pasley (2002), S.158ff.

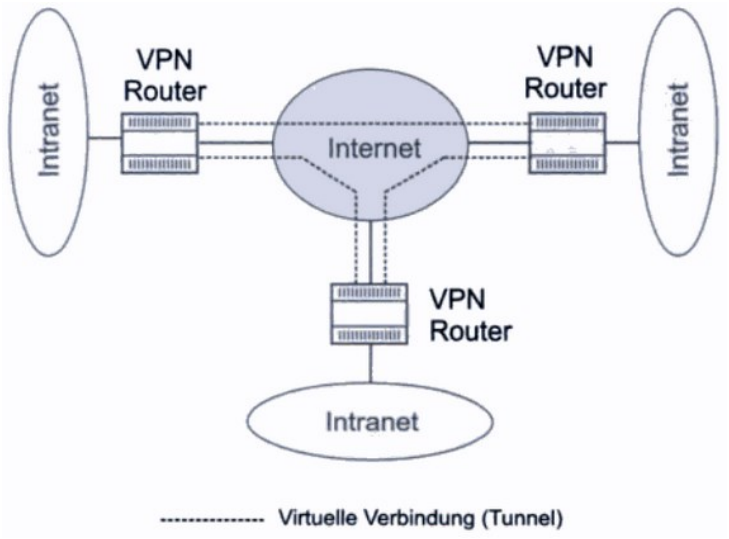


Abbildung 9: Branch-Office VPN, Quelle: Lipp (2001), S.44.

2.2.3 Extranet VPN

Mit dem Extranet VPN wird beabsichtigt, nur bestimmte Daten für bestimmte Zielgruppen zu veröffentlichen bzw. diese auszutauschen (siehe Abbildung 10). Ein gutes Beispiel hierfür ist ein Kunden- oder Lieferantenportal (z.B. um Lieferzeiten oder Lagerbestände zu aktualisieren). Diese Daten müssen vertraulich behandelt werden. Eine Authentifizierung mittels Benutzernamen und Passwort ist daher obligatorisch. Außerdem muss eine Vereinbarung über die gemeinsam verwendete VPN-Lösung aufgestellt und von beiden Parteien akzeptiert werden.[23]

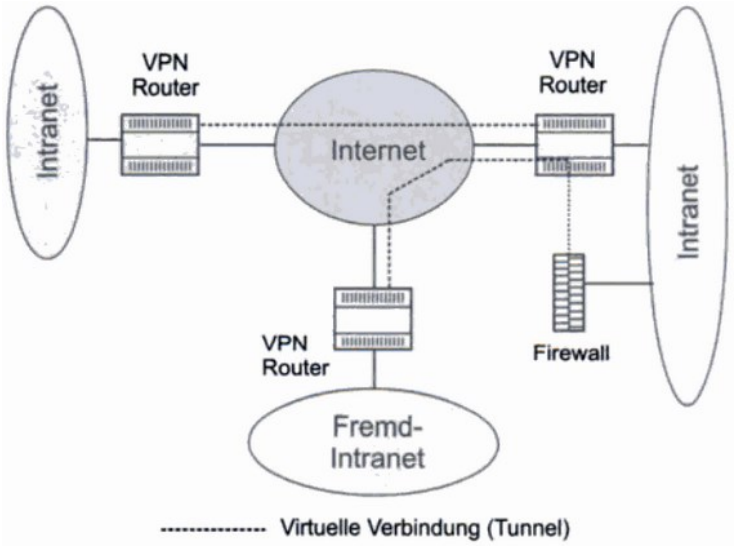


Abbildung 10: Extranet VPN, Quelle: Hekerens (2001), Online-Quelle [1.Juli.2016].

[23] Vgl. Pasley (2002), S.160ff.

2.3 VPN Topologien

VPN-Topologien legen fest, welche Netzwerkkomponenten ihren Einsatz finden und wie diese miteinander verbunden sind.[24] Die verwendete Topologie hängt von der Struktur der Organisation ab und welche Anforderungen an die VPN-Verbindung gestellt werden.[25] Dabei werden drei verschiedene Topologien unterschieden.

2.3.1 Hub-and-Spoke Topologie

In einer Hub-and-Spoke Topologie sind mehrere entfernte Geräte (Spokes) mit dem zentralen Gerät (Hub) verbunden. Durch den sicheren Tunnel zwischen jedem einzelnen „Spoke“ und dem „Hub“ (siehe Abbildung 11) können Daten untereinander sicher ausgetauscht werden.[26] Der Name entstand durch die Anlehnung an ein Rad. Der „Hub“ stellt die Nabe eines Rades dar, die „Spokes“, oder auch Teilnehmer, die einzelnen Speichen. Die meisten VPN-Verbindungen (Branch-Office VPN) werden durch diese Topologie realisiert.[27]

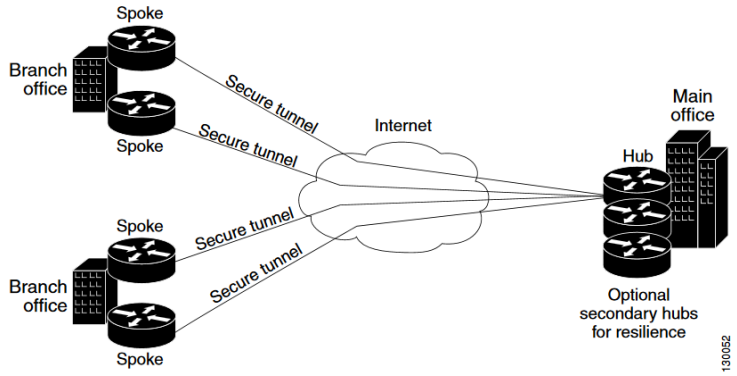


Abbildung 11: Hub-and-Spoke Topologie, Quelle: Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.312.

2.3.2 Point-to-Point Topologie

Bei der Point-to-Point Topologie findet die Kommunikation direkt zwischen zwei Endgeräten statt.[28]

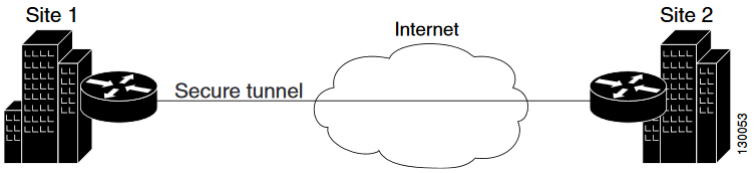


Abbildung 12: Point-to-Point Topologie, Quelle: Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.313.

[24] Vgl. Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.312.

[25] Vgl. Guichard/Pepelnjak (2001), S.129.

[26] Vgl. Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.312.

[27] Vgl. Schäfer/Roßberg (2014), S.597ff.

[28] Vgl. Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.313.

2.3.3 Full-Mesh Topologie

Die zuletzt erläuterte Topologie wird verwendet, um mehrere Endgeräte untereinander zu verbinden. Das heißt eine Kommunikation mit jedem einzelnen Gerät ist über einen sicheren Tunnel möglich. Durch die sich aus den vielen Verbindungen unter den Geräten ergebende Redundanz ergibt sich automatisch eine hohe Verfügbarkeit dieser VPN-Lösung. Sollte es zu einem Ausfall einer Verbindung kommen, werden andere Geräte dadurch nicht beeinträchtigt und können unter Umständen auch einen anderen Verbindungsweg wählen.[29]

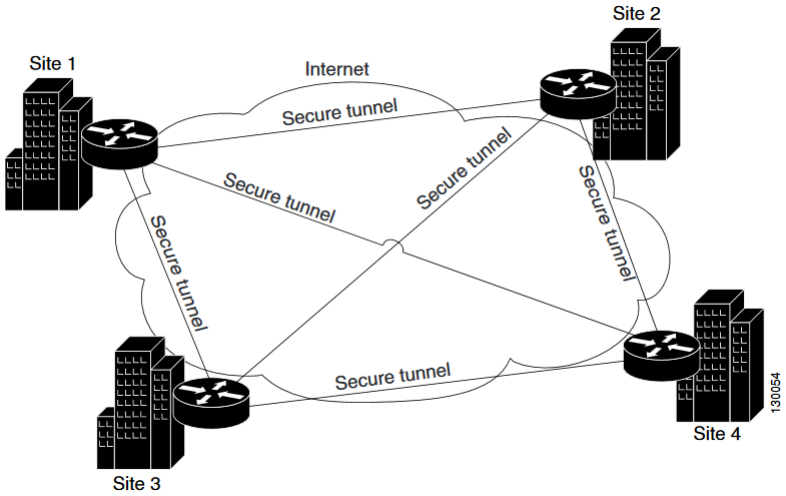


Abbildung 13: Full-Mesh Topologie, Quelle: Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.314.

[29] Vgl. Cisco Systems, Inc. (2009), Online-Quelle [11.Juli.2016], S.314.

3 VPN-PROTOKOLLE

Um Daten von Rechner A zu Rechner B übertragen zu können, bedarf es unterschiedlicher Dienstleistungen auf unterschiedlichen „Schichten“ (englisch: Layers), welche im ISO/OSI-Schichtenmodell (siehe Abbildung 14) definiert sind. Eine Kommunikation findet dadurch statt, dass jede Schicht der übergeordneten Dienstleistungen zur Verfügung stellt, ohne dabei die Daten bei der Übernahme auf Richtigkeit zu überprüfen.[30] Das Modell wird in sieben Schichten eingeteilt, wobei die Schichten 1-4 transportorientiert, und die Sichten 5-7 anwendungsorientiert sind.[31]

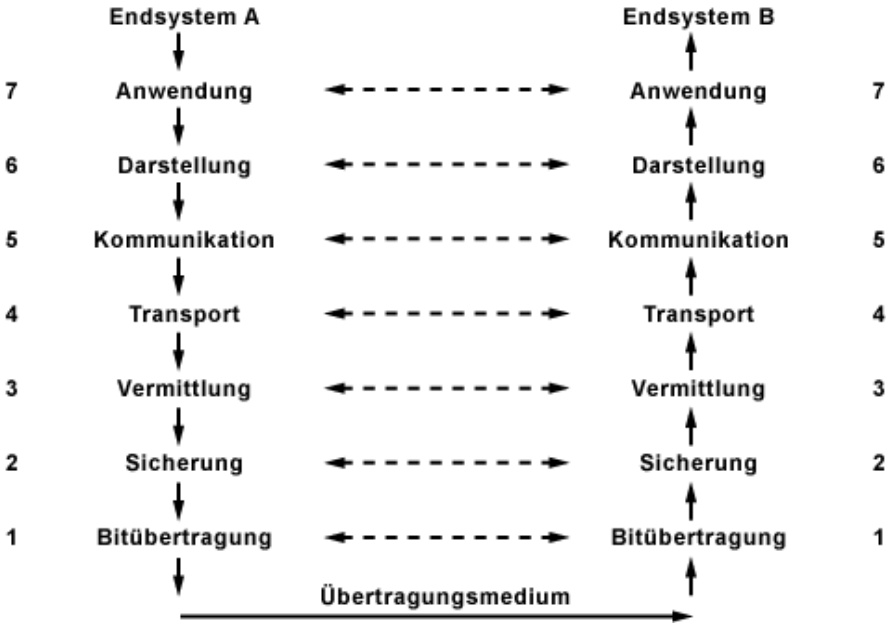


Abbildung 14: OSI-Modell, Quelle: Elektronik Kompendium (2016), Online-Quelle [12.Juli.2016].

Um die Funktionsweise dieses Modells verständlicher darzustellen, werden nun in den folgenden Absätzen die einzelnen Schichten und deren Aufgaben näher erläutert.

Schicht 1, auch Übertragungsschicht (engl. **Physical Layer**) genannt, ist verantwortlich für die „*eigentliche physikalische Übertragung in Form eines transparenten Bitstromes*“[32]. Hier werden sowohl das Übertragungsmedium als auch das physikalische Umfeld für die Übertragung definiert. Beispiele für Geräte, die in dieser Schicht für die wesentlichen Übermittlungsdienste benötigt werden, sind Repeater, Modems oder Transceiver.

Die darüber liegende Schicht 2, auch als **Data Link Layer** bezeichnet, unterteilt die Bits in Datenpakte und nimmt auch eine Fehlererkennung vor. Sogenannte Bridges finden ihren Einsatz in dieser Schicht. Diese leiten die empfangenen Datenpakete auf einem bestimmten Weg weiter. Des Weiteren können noch Filter eingestellt werden, die durch Bitmuster realisiert werden. Bei Übereinstimmung mit dem Bitmuster entscheidet die Bridge über Weiterleitung oder Verwerfung des Paketes.

[30] Vgl. Hein (2000), S.21.

[31] Vgl. Elektronik Kompendium (2016), Online-Quelle [12.Juli.2016].

[32] Hein (2000), S.33.

In der Schicht 3 (**Network Layer**) wird die Funktion der Wegefindung (Routing) übernommen, welche es ermöglicht, mehrere Netzwerke miteinander zu verbinden. Das wohl am häufigsten eingesetzte Gerät in der Netzwerk- oder Vermittlungsschicht ist neben Gateway und Vermittlungsknoten der Router, deren Arbeitsweise vom jeweiligen Protokoll abhängig ist. Das heißt, ein IP-Router ist für alle Nicht-IP-Protokolle (z.B. IPX oder DECnet) nicht durchlässig. Die Hauptaufgabe des Routers besteht zum einen im Aus- und Wiederverpacken der Pakete, die dabei mit den netzspezifischen Protokollinformationen ausgestattet werden, und zum anderen in der Wegefindung in einem Netzwerk.

Die nächst höhere Schicht 4 (**Transport Layer**) führt die tatsächliche Datenübertragung zwischen zwei Geräten durch. Dabei können mehrere Protokolle diese Aufgabe übernehmen. Man unterscheidet zwischen verbindungsorientierten (z.B. TCP[33]) und verbindungslosen (z.B. UDP[34]) Protokollen. Der Unterschied liegt in der Überprüfung, ob das Paket am Empfänger angekommen ist. TCP kann somit eine sichere Übertragung garantieren, während UDP keine Überprüfung vornimmt. Dies führt bei verbindungslosen Protokollen allerdings auch zu einer besseren Performance.[35]

Die Anwendungsschichten des OSI-Referenzmodells bilden[36]:

- Schicht 5 (**Session Layer**): Hier findet die Prozesskommunikation statt und es werden die ausgetauschten Informationen umgesetzt und dargestellt.
- Schicht 6 (**Presentation Layer**): Auf dieser Ebene werden die Daten für das jeweilige System codiert bzw. decodiert.
- Schicht 7 (**Application Layer**): Je nach Anwendung finden hier verschiedenste Protokolle (z.B. SMTP[37], FTP[38], etc.) ihren Einsatz. Auf dieser Schicht arbeitende Geräte werden als Gateways bezeichnet.

Kommunizieren nun zwei einander entsprechende Schichten miteinander, müssen festgelegte Regeln eingehalten werden. Hierbei wird von einem sogenannten Protokoll gesprochen, welche VPN-Verbindungen erst ermöglichen.[39] Dabei werden diese auch nach den Schichten des OSI-Referenzmodells eingeteilt.

3.1 Protokolle auf ISO/OSI-Schicht 2

Im folgenden Kapitel werden Protokolle auf der Sicherungsschicht (Data Link Layer) behandelt. Das auf dieser Schicht arbeitende PPP (Point-to-Point Protocol) spielt für die nachfolgend erläuterten Protokolle

[33] TCP…Transmission Control Protocol.

[34] UDP…User Datagram Protocol.

[35] Vgl. Hein (2000), S.33ff.

[36] Vgl. Hein (2000), S.37.

[37] SMTP…Simple Mail Transfer Protocol.

[38] FTP…File Transfer Protocol.

[39] Vgl. Hein (2000), S.21.

eine wichtige Rolle. Es ist „*ein Verfahren zum Transport von Netzwerkprotokollen über Punkt-zu-Punkt-Verbindungen“*[40]. Durch PPP-Verbindungen werden häufig Clients mit Remote Host verbunden. Das PPP hat die Aufgabe, IP-Pakete oder andere Protokolle zu transportieren. Die Protokolle PPTP, L2TP und L2F werden alle für die Bildung eines Tunnels durch das Internet verwendet, welcher die Punkt-zu-Punkt-Verbindung ersetzt. Durch die Nutzung der PPP-Standards, lassen sich auch Vorteile (wie z.B. Benutzerauthentifizierung oder dynamische Adresszuteilung) nutzen.

3.1.1 PPTP

Das Point-to-Point Tunneling Protocol (PPTP) zählt zu den nicht standardisierten Protokollen, da es nicht durch ein Standardisierungsgremium zu einem Standard verabschiedet wurde. Hingegen wurde es von einer Reihe von Firmen (Ascend Communications, U.S. Robotics, 3Com Corporation, Microsoft Corporation und ECI Telematics) entwickelt. Daraus ergab sich auch die Bildung des PPTP-Forums.[41]

Als ein VPN-Tunneling-Protokoll ist die Aufgabe von PPTP, Pakete in einen IP-Rahmen zu verschachteln. Dies ermöglicht ein Routing in Medien, in welchen nur IP-Pakete verstanden werden (z.B. Internet). Dieser Verkapselungsmechanismus basiert auf dem GRE (Generic Routing Encapsulation) Standard, welcher für das Tunneling von Protokollen im Internet verwendet wird.

Wie in Abbildung 15 ersichtlich, besteht ein PPTP Paket aus dem Delivery Header, dem IP Header, dem GREv2 Header und dem Payload Datagram. Der Delivery Header gibt Auskunft über das Übertragungsmedium (z.B. Ethernet, PPP, etc.). Der IP-Header beinhaltet IP-spezifische Informationen, wie z.B. die Paketlänge, Quell- und Zieladresse. Der GREv2-Header enthält einerseits Informationen zum verkapselten Paket, und andererseits zu PPTP-relevanten Daten, die die Verbindung zwischen dem Server und dem Client betreffen. Das Payload Datagram beinhaltet schlussendlich die eigentlichen Daten des verkapselten Paketes.[42]

Abbildung 15: Aufbau eines PPTP Pakets, Quelle: In Anlehnung an: Scott/Wolfe/Erwin (1999), S.70.

Um möglichst große Sicherheit von PPTP zu gewährleisten werden Authentifizierungs- und Verschlüsselungsmethoden angewandt. PPTP verwendet das Windows NT RAS Authentifizierungsverfahren, welches verschiedene Optionen (je nach gewünschter Sicherheit) zulässt[43]:

- „*Accept encrypted authentication*“

[40] Lipp (2001), S.295.

[41] Vgl. Lipp (2001), S.180.

[42] Vgl. Scott/Wolfe/Erwin (1999), S.70.

[43] Scott/Wolfe/Erwin (1999), S.74f, vgl. Scott/Wolfe/Erwin (1999), S.74f.

- Verwendung des Internet Authentifizierungsstandards CHAP[44], welches sicherstellt, dass das Passwort zwischen Client und Server nie unverschlüsselt übertragen wird.

- „*Accept Microsoft encrypted authentication*“
  - Verwendung von MS-CHAP, welches auf dem Standard CHAP basiert. Nachteilig ist hier, dass der Einsatz von MS-CHAP nur auf bestimmten Plattformen möglich ist.

- „*Accept any authentication, including clear text*“
  - Für die Authentifizierung werden CHAP, MS-CHAP und PAP[45] akzeptiert. Bei PAP wird das Passwort allerdings in Klartext übertragen.

Die zuvor erwähnten Maßnahmen zur Sicherheitserhöhung sind allerdings nicht mit denen des IPSec-Protokolls (siehe Kapitel 3.2 Protokoll auf ISO/OSI-Schicht 3 – IPSec) zu vergleichen. Durch die Ableitung des Schlüssels (zur Datenverschlüsselung) aus dem Benutzerpasswort, ließen sich in der Vergangenheit mehrere Angriffe durchführen.[46]

Abschließend ist zu sagen, dass PPTP durch die einfache Einrichtung und Verfügbarkeit auf vielen Endgeräten zwar Vorteile bringt, durch die Sicherheitslücken allerdings für eine sichere VPN-Verbindung nur bedingt einsetzbar ist.[47]

3.1.2 L2TP

Das Layer 2 Tunneling Protocol (L2TP) zählt zu den standardisierten Protokollen. Es wurde auf Basis der nicht standardisierten Protokolle von PPTP und L2F (Layer 2 Forwarding) weiterentwickelt. Die Aufgabe des L2TP ist es, die Kommunikation über PPP auch über andere als nur Punkt-zu-Punkt-Verbindungen zu ermöglichen.[48]

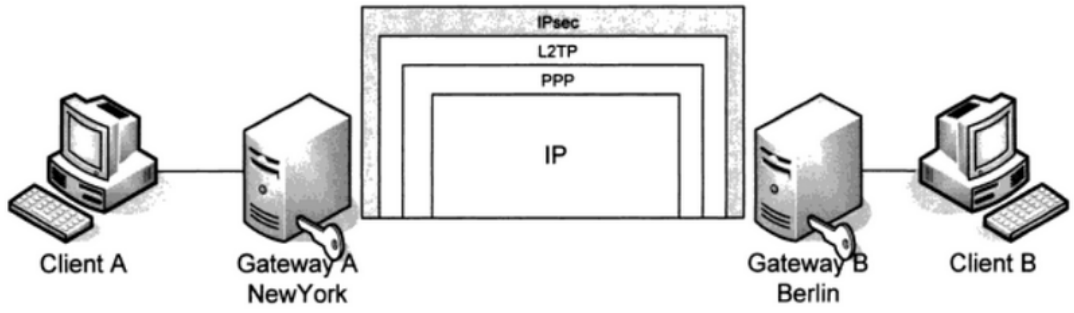


Abbildung 16: IPSec als Schutz für den L2TP-Tunnel, Quelle: Spenneberg (2010), S.118.

[44] CHAP…Challenge Handschake Authentication Protocol.

[45] PAP…Password Authentication Protocol.

[46] Vgl. Lipp (2001), S.180.

[47] Vgl. Kompendium (2016), Online-Quelle [15.Juli.2016].

[48] Vgl. Lipp (2001), S.295.

Beim alleinigen Einsatz des Protokolls können mehrere Sicherheitsaspekte wie Authentizität, Integrität und Vertraulichkeit der Daten nicht garantiert werden. In Kombination mit dem IPSec-Protokoll allerdings (baut eine verschlüsselte Verbindung zwischen den Endgeräten auf), ist dies jedoch möglich (siehe Abbildung 16).

Aufbauend auf die IPSec-Verbindung zwischen den Kommunikationspartnern wird ein L2TP-Tunnel aufgebaut, welcher folgende Vorteile mit sich bringt[49]:

- Beim Aufbau des Tunnels wird eine erneute Benutzerauthentifizierung angefordert, da diese von der IPSec-Authentifizierung abweichen kann
- Die Endpunkte der beiden Tunnels können abweichend sein
- Auch Nicht-IP-Pakete (z.B. IPX[50]) können transportiert werden
- Die Verwendung anderer IP-Adressen als die IPSec-Verbindung ist möglich

Zwei Nachrichtenformen werden für das L2TP-Protokoll verwendet. Steuerungsnachrichten bauen Tunnel auf, verwalten und löschen diese. Dabei wird die erfolgreiche Datenübertragung garantiert. Im Gegensatz zu den Datennachrichten. Diese werden für den Tunneltransport der PPP-gekapselten Daten verwendet.[51]

Zusammenfassend ist L2TP in Kombination mit IPSec gut dafür geeignet, eine sichere Verbindung von außen zu einem privaten Netzwerk herzustellen.

3.1.3 L2F

Das Layer 2 Forwarding Protokoll ist ein von Cisco entwickeltes Tunneling-Protokoll, welches auch (wie oben beschrieben) einen Teil des L2TP darstellt. Der Unterschied zum gleichzeitig entwickelten PPTP ist, dass keine VPN-Client-Software notwendig ist, da alle Dienste von einer speziellen L2F-Hardware zur Verfügung gestellt werden. Zwei Server sind dabei vorgesehen – ein Server beim Internet Service Provider (ISP) und einer am Unternehmensstandort. Der Tunnel wird dann vom Server beim ISP aufgebaut.[52]

Da das Protokoll heutzutage in dieser Form nicht mehr eingesetzt wird (sondern nur mehr in Teilen des Nachfolgers L2TP), wird es in dieser Arbeit nicht näher erläutert.

3.2 Protokoll auf ISO/OSI-Schicht 3 – IPSec

Das IPSec (IP-secure) Protokoll operiert auf Schicht 3 (Network Layer) des OSI-Referenzmodells und ist ein standardisiertes Sicherheitsprotokoll. Es entstand aus der Idee heraus, Schwachstellen von IP durch IPv6 zu beheben. Durch die schleppende Einführung von IPv6 entdeckte man, dass die

[49] Vgl. Spenneberg (2010), S.118.

[50] IPX…Internetwork Packet Exchange.

[51] Vgl. Spenneberg (2010), S.119.

[52] Vgl. Beasley/Nilkaew (2012), S.488.

Sicherheitsvorkehrungen, die eigentlich für IPv6 gedacht waren, auch am IPv4 angewendet werden können. Als IPSec werden nun genau diese Sicherheitsvorkehrungen bezeichnet. Da das Protokoll bei der ursprünglichen Konzipierung für eine andere Umgebung gedacht war, gibt es Schwachpunkte, die beim Einsatz auf IPv4 beachtet werden müssen (z.B. die Ver- und Entschlüsselung muss auf der Firewall erfolgen).[53] Die Spezifikationen zu IPSec wurden aus einer Reihe von unterschiedlichen Dokumenten zusammengefasst und als Request For Comments veröffentlicht.

3.2.1 IPSec – Sicherheitsaspekte

Als Motivation bei der Entwicklung stand die Standardisierung von Layer 3-Zusatzfunktionen für eine umfassende Sicherheit (Authentifizierung, Datenintegrität und Verschlüsselungsmechanismen) im Vordergrund. IPSec stellt nämlich eine Verschlüsselung und die Authentifikation auf Schicht 3 bereit, die somit für eine Ende-zu-Ende-Sicherheit sorgt. Dies ermöglicht, dass Endgeräte (und ihre Applikationen) nicht geändert werden müssen, da die verschlüsselten IPSec-Pakete wie IP-Pakete über jedes IP-Netzwerk (z.B. Internet) geroutet werden können.[54]

IPSec stellt unterschiedliche Optionen zum Thema Verschlüsselung und Authentifikation bereit. Jede IPSec-Verbindung kann sich nach dem Grad der Sicherheit unterscheiden. Der Ablauf ist dabei folgender[55]:

- Zwischen den zwei Knoten findet eine Aushandlung über den Grad der Sicherheit statt
    - Sicherheitsmechanismen werden festgelegt (Verschlüsselung: DES[56] oder IDEA[57], Integrität: MD5[58] oder SHA[59])
- Session-Keys werden ausgetauscht
- Ergebnis: Security Association (SA) beschreibt, wie für die Sicherheit zwischen den beiden Knoten gesorgt wird und stellt jeweils nur eine unidirektionale Verbindung dar (d.h. bei Kommunikation von zwei Knoten ergeben sich zwei SAs – von A nach B und von B nach A). Die Kennzeichnung erfolgt über den Security Parameter Index (SPI) und die IP-Zieladresse.

Für die Aushandlung der Sicherheitsparameter ist das Internet Key Management Protocol (IKMP) verantwortlich. Konkret wird hier der ISAKMP[60]/Oakley-Mechanismus verwendet, welcher einen authentifizierten und sicheren Tunnel zwischen den zwei Knoten aufbaut. Die Sicherheitsparameter werden dann über diesen Tunnel ausgehandelt. Voraussetzung dafür ist allerdings, dass sich die beiden

[53] Vgl. Reiser (2000), S.86f.

[54] Vgl. Hein (2000), S.989.

[55] Vgl. Hein (2000), S.989.

[56] DES…Data Encryption Standard.

[57] IDEA…International Data Encryption Algorithm.

[58] MD5…Message-Digest Algorithm 5.

[59] SHA…Secure Hash Algorithm.

[60] ISAKMP…Internet Security Association and Key Management Protocol.

Knoten gegenseitig authentifizieren und gemeinsame Schlüssel austauschen. Der Oakley-Standard stellt dabei folgende Authentifizierungsmechanismen zur Verfügung[61]:

- Pre-shared Keys
    - Ein Schlüssel wird auf den Rechnern erzeugt. Die Schlüsselsequenz (verschlüsselte Daten) wird dem jeweiligen Partnern zugesendet. Kann dieser die Daten entschlüsseln (impliziert die Verwendung des gleichen Schlüssels), gelten die beiden Partner als authentifiziert.
- Public Key Cryptography
    - Zufallszahlen werden auf den Rechnern generiert und mit dem öffentlichen Schlüssel des Partners verschlüsselt. Die Prüfung erfolgt durch Erzeugung einer Schlüsselsequenz vom Partner (die die Zufallszahl beinhaltet), die wiederum mit dem privaten Schlüssel entschlüsselt wird. Sind die Zufallszahlen ident, ist der Partner authentifiziert.
- Digital Signature
    - Ein Datensatz wird durch einen Hash-Algorithmus verschlüsselt. Dieser „Fingerabdruck“ wird dem Datensatz angehängt. Der Partner vergleicht beim Empfang den Fingerabdruck mit dem Hash des empfangenen Datensatzes. Sind die beiden identisch, erfolgt eine Authentifizierung.

Nach Authentifizierung der beiden Knoten muss der Tunnel, über den die Aushandlung der Sicherheitsparameter stattfindet, ebenfalls verschlüsselt werden. Dazu ist es notwendig, dass beide Partner einen gemeinsamen Session-Schlüssel besitzen. Dieser Schlüssel wird durch das Diffie-Hellman-Protokoll festgelegt. Nach sicherem Aufbau des Tunnels (Verschlüsselung) findet nun die Aushandlung der Sicherheitsparameter statt. Dabei werden dem Kommunikationspartner einige Parameter angeboten. Der Gegenüberliegende wählt danach jene Sicherheitsmechanismen aus, die für die Verbindung gelten sollen. Nach Einigung müssen nun nur noch die Schlüssel für AH (Authentication Header) bzw. ESP (Encapsulation-Security-Payload) erstellt und ausgetauscht (wieder über das Diffie-Hellman-Protokoll) werden. Eine vollständige IPSec Security Association (SA) ist nach Beendigung dieses Prozesses aufgebaut.[62]

3.2.2 IPSec – Protokollaufbau

Im IPSec-Protokoll sind die Sicherheitsprotokolle bzw. –funktionen AH und ESP integriert. Der IP-Authentication-Header übernimmt die Authentifizierungs- und Integritätsfunktion, während das Encapsulation-Security-Payload die Verschlüsselung ermöglicht. Beide Protokolle können sowohl im Transport- als auch im Tunnelmodus ausgeführt werden. Der Transportmodus sorgt sowohl in der IP-Ebene als auch auf höheren Schichten für Sicherheit und kann nur zwischen zwei Endgeräten eingesetzt werden. Außerdem ist die Bandbreitenauslastung im Transportmodus niedriger als im Tunnelmodus. Im

[61] Vgl. Hein (2000), S.990.

[62] Vgl. Hein (2000), S.990.

Tunnelmodus wird der Fokus auf die Sicherheit des IP-Paketes gelegt. Hierfür wird ein IPSec-Header zwischen dem inneren und äußeren IP-Header eingefügt.[63]

3.2.2.1 Authentication Header

Der AH ist als sogenannte „Zwischenschicht" zwischen den IP-Headern und den Transport-Layer-Headern einsetzbar. Durch den AH wird sichergestellt, dass die gesendeten Daten während der Übertragung unverändert bleiben (Integritätsfunktion). Des Weiteren wird auch, wie oben beschrieben, die Authentifizierungsfunktion durch den AH durchgeführt. Dafür werden die zwei Methoden MD5 und SHA verwendet.

Im Transport-Modus wird der AH nach dem IP-Header eingefügt (siehe Abbildung 17) und sorgt damit auch für die Sicherheit von höheren Protokollen (wie z.B. TCP, UDP oder ICMP[64]).

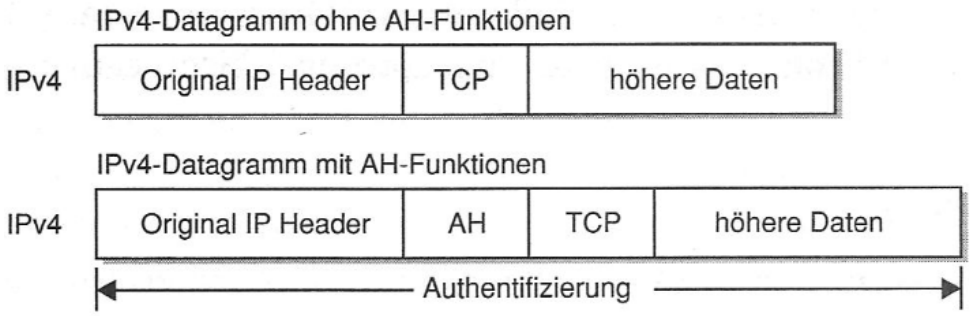


Abbildung 17: AH-Header im Transport-Modus, Quelle: Hein (2000), S.992.

Zum Schutz des gesamten IP-Headers wird im Tunnel-Modus der AH zwischen den „neuen" (enthält die IP-Adressen der Tunnelendpunkte) und den „originalen" IP-Header (enthält die Quell- und Zieladressen der kommunizierenden Geräte) eingefügt (siehe Abbildung 18).[65]

Abbildung 18: AH-Header im Tunnel-Modus, Quelle: Hein (2000), S.992.

3.2.2.2 Encapsulation-Security-Payload

Durch ESP ist es möglich, die IP-Daten zu verschlüsseln. Wie auch der AH wird der ESP-Header im Transport-Modus nach dem IP-Header eingefügt (siehe Abbildung 19). Ebenso wird auch im Tunnel-Modus der gesamte IP-Header durch den ESP-Mechanismus geschützt (siehe Abbildung 20).[66]

[63] Vgl. Hein (2000), S.991, vgl. Doraswamy/Harkins (2003), S.45.

[64] ICMP…Internet Control Message Protocol.

[65] Vgl. Hein (2000), S.991f.

[66] Vgl. Hein (2000), S.992ff.

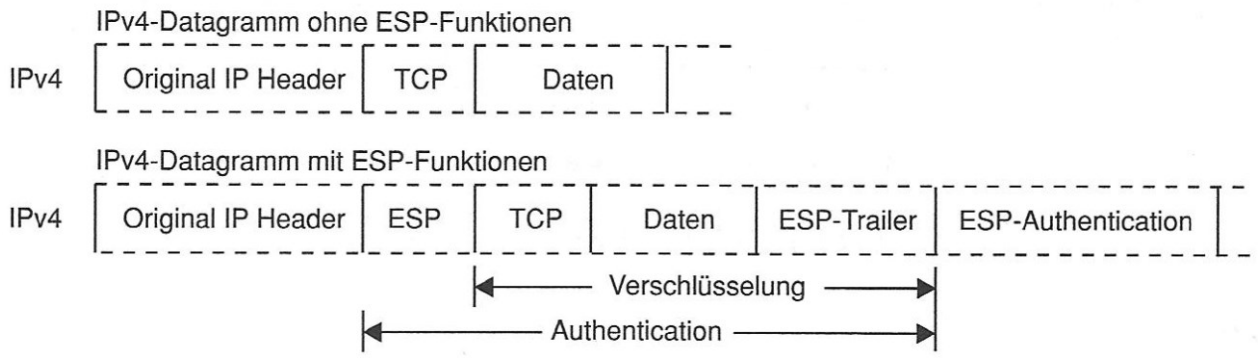


Abbildung 19: ESP-Header im Transport-Modus, Quelle: Hein (2000), S.993.

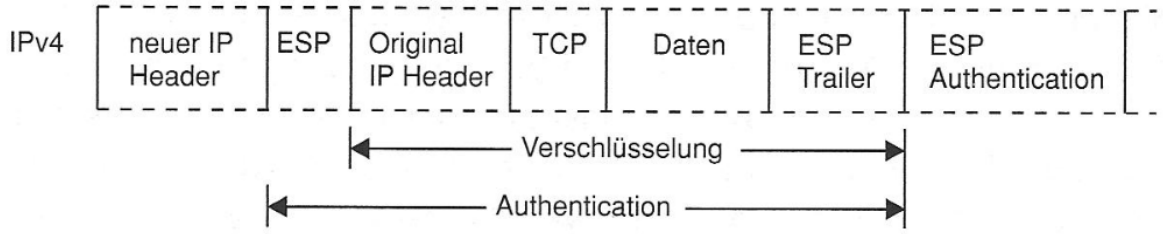


Abbildung 20: ESP-Header im Tunnel-Modus, Quelle: Hein (2000), S.994.

3.2.3 IPSec – Funktionsweise

Um die Funktionsweise von IPSec näher zu erläutern, wird im folgenden Kapitel der Ablauf einer sicheren Kommunikation zwischen zwei Endgeräten über ein IP-Netzwerk näher erklärt und in Abbildung 21 dargestellt.[67]:

1. In Endsystem 1 (ES1) wird eine Nachricht (IP-Datenpaket) erzeugt
2. Durch Prüfung des IPSec-Stacks kann ermittelt werden, ob die Nachricht an Endsystem 2 (ES2) verschlüsselt übermittelt werden soll. Die Auskunft der Security Police Database (SPD1)[68] durch ES1 gibt Aufschluss über die Sicherheitsmechanismen, die das Paket sichern sollen
3. Ein Eintrag in der SPD1 wird erstellt und eine SA1 generiert (durch IKE[69]1), sofern diese noch nicht vorhanden
4. Aushandlung der Sicherheitsparameter zwischen IKE1 und IKE2
5. Erzeugung der SA und Speicherung in SPD1 (die SA ist nun auf beiden Seiten bekannt)
6. Mitteilung durch SPD1 an ES1, dass Paket gesichert übermittelt werden muss und stellt dabei SA1 bereit
7. Mithilfe von SA1 wird das Paket gesichert (z.B. verschlüsselt)
8. Übermittlung des Datenpakets an ES2

[67] Vgl. Stark (2001), S.78.

[68] SPD…Definiert, wie IP-Pakete von IPSec verarbeitet werden sollen.

[69] IKE…Internet Key Exchange

9. Prüfung durch ES2, ob IPSec-Header vorhanden ist und eine SA bereits existiert (SA2 wurde in Schritt 5 bereits erstellt)
10. Entsicherung (z.B. Entschlüsselung) des Datenpakets durch SA2
11. Abschluss der sicheren Übertragung von ES1 nach ES2 (Paket kann nun von ES2 weiterverarbeitet werden)

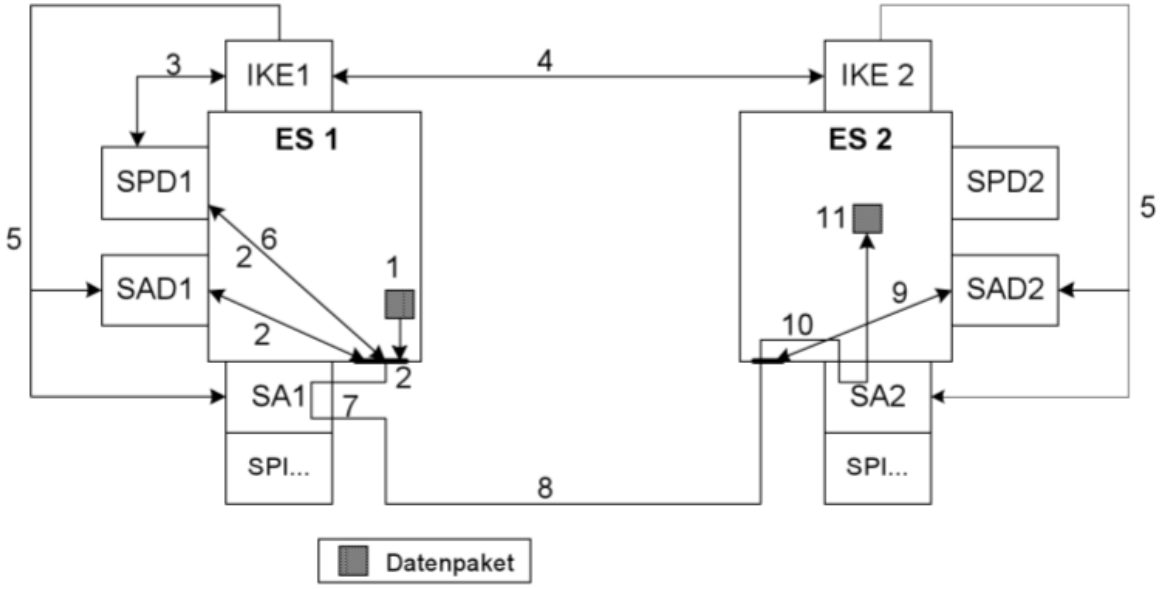


Abbildung 21: Ablauf einer sicheren Übertragung, Quelle: Stark (2001), S.78.

Zusammenfassend ist IPSec derzeit eines der am häufigsten eingesetzten Protokolle für VPNs, welches sowohl die Authentifizierung und das Schlüsselmanagement (durch IKE) bereitstellt, als auch die Datenverschlüsselung (stellt Vertraulichkeit und Integrität durch ESP und AH sicher).[70]

3.3 Protokolle auf höheren ISO/OSI-Schichten

Nach Abschluss der Protokolle auf Schicht 2 und 3 setzt das folgende Kapitel den Fokus auf höhere Schichten. Dort arbeiten z.B. Protokolle wie SSL (Secure Sockets Layer) oder TLS (Transport Layer Security).

3.3.1 SSL

Die Entwicklung des Secure Sockets Layer Protokolls durch die Firma Netscape Communications begann im November 1993. Im ungefähren Abstand von sechs Monaten wurden zwei weitere Versionen entwickelt. Die Version SSL 3.0 (Veröffentlichung 1995) ist der aktuellste Stand des Protokolls.

Zu Beginn der Entwicklungen wurde eigentlich an einem Protokoll gearbeitet, das den Fokus auf Sicherheit legt. Im Laufe der Zeit entwickelte sich daraus allerdings eine ganze Schicht (siehe Abbildung 22: SSL als Protokollschicht, Quelle: In Anlehnung an: Thomas (2000), S.8.), die ebenfalls nur für die Sicherheit von Web-Verbindungen verantwortlich ist. Der Einfluss auf die darüber- und darunterliegenden

[70] Vgl. Mel/Baker (2001), S.246.

Schichten ist gering. Die Schnittstellen von HTTP-Applikationen arbeiten mit SSL sehr ähnlich wie mit TCP (ohne SSL-Schicht). Und für TCP stellt das darüber liegende SSL-Protokoll nur eine weitere Applikation dar, die den Service von TCP nutzt.

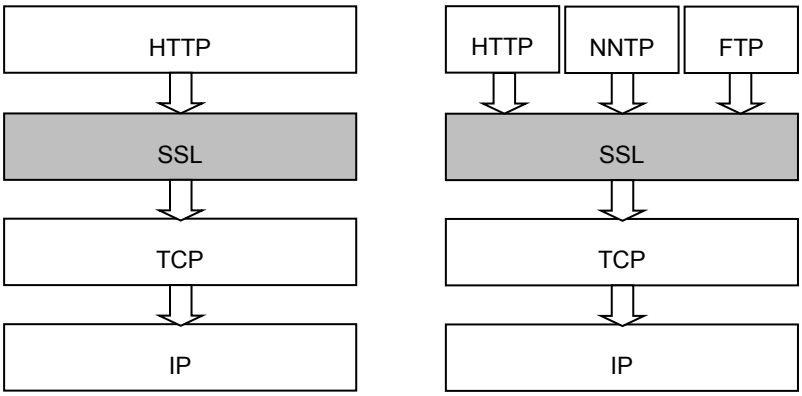


Abbildung 22: SSL als Protokollschicht, Quelle: In Anlehnung an: Thomas (2000), S.8.

Ein großer Vorteil, den SSL mit sich bringt, ist, dass über der SSL-Schicht weitere Applikationen (neben HTTP) verwendet werden können (z.B. Net News Transfer Protocol oder File Transfer Protocol).[71]

Wie jede Technologie hat auch SSL Einschränkungen hinzunehmen. Da es bei SSL im Speziellen um die Sicherheit geht, ist in diesem Fall besonders darauf Acht zu geben. Die drei Einschränkungsbereiche sind folgende[72]:

- Fundamentale Einschränkungen des Protokolls selbst
    - Ergeben sich aus dem Design und der Anwendung des Protokolls (z.B. SSL kann nicht in Kombination mit UDP (da es zu unzuverlässig ist) eingesetzt werden).
- Einschränkungen durch Tools, auf denen SSL basiert
    - Da SSL für sich alleine nur ein Kommunikationsprotokoll darstellt, muss es zusammen mit anderen Komponenten eingesetzt werden, die z.B. für den Verschlüsselungs- oder Signaturalgorithmus verantwortlich sind. Dabei ist darauf zu achten, dass es sich hier um sehr sichere Algorithmen handelt, da SSL diese nutzt und deren Schwächen nicht ausgleichen kann.
- Einschränkungen durch die Umgebung
    - SSL alleine kann kein ganzes Netzwerk schützen. Die Sicherheit hängt auch von der eingesetzten Hardware, deren Betriebssystemen und natürlich von den Anwendern ab.

Wie nun eine SSL-Session im Detail funktioniert, wird in Tabelle 3 näher erläutert. Bob stellt einen Kunden dar, der mit Alice, einer Händlerin, Kontakt aufnehmen möchte.

Vorschlag von Parametern	Bob sendet an Alice eine Nachricht im Klartext und schlägt Parameter

[71] Vgl. Thomas (2000), S.4ff.

[72] Vgl. Thomas (2000), S.12ff.

	für die Konversation vor (z.B. Version von SSL, Verfahren zum Schlüsselaustausch, etc.)
Festlegung von Parametern	Alice antwortet auf die Nachricht und teilt ihre Wahl der Parameter Bob mit.
Austausch der Zertifikate	Alice sendet an Bob ihr digitales Zertifikat. Bob überprüft dies mit einem Zertifikat, das von einer CA[73] (vertrauenswürdigen Zertifizierungsstelle) herausgegeben wurde.
Schlüsselvereinbarung	Bob generiert eine 48-byte Zufallszahl (auch „Pre-Master Secret“ genannt), welche er dann mit dem öffentlichen Schlüssel von Alice aus dem Zertifikat verschlüsselt und an Alice sendet. Alice entschlüsselt diese wiederum mit ihrem privaten Schlüssel.
Erstellung des „Master Secret“	Aus dem „Pre-Master Secret“ wird mithilfe diverser Verschlüsselungsoperationen der „Master Secret“ erstellt, aus dem die Schlüsseldaten für die Konversation abgeleitet werden können.
Authentifizierung	Bob sendet an Alice eine Nachricht, die er mit den vereinbarten Schlüsseln verschlüsselt. Diese sogenannte „Finished Handshake“-Nachricht, ist die erste Nachricht, die mit den vereinbarten Schlüsseln verschlüsselt wird. Kann nun Alice diesen gesendeten Text entschlüsseln, kann Bob davon ausgehen, dass auch Alice die richtigen Schlüsseldaten verwendet.

Tabelle 3: SSL-Session im Detail, Quelle: vgl. Mel/Baker (2001), S.218ff.

Der Verbindungsaufbau bei SSL ist ähnlich einem IPSec-Remote-Access. Eine sichere Verbindung von einem Rechner zu einem SSL-VPN-Gateway wird aufgebaut (siehe Abbildung 23). Dieser wiederum ermöglicht den Zugriff auf unternehmensinterne Applikationen oder Daten. Der Unterschied zur IPSec-Verbindung liegt allerdings in der Anzahl der Verbindungen. Während bei IPSec nur eine einzige Verbindung für alle Daten besteht, wird bei SSL für jede vom Client verwendete Applikation eine eigene Verbindung aufgebaut.

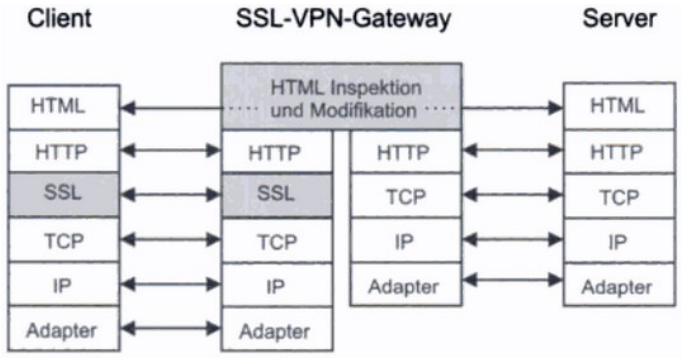


Abbildung 23: SSL-VPN-Aufbau, Quelle: Lipp (2001), S.283.

[73] CA…Certification Authority.

Der Aufbau eines sicheren Tunnels ist durch die Verwendung des „Secure Socket Tunneling Protocol“ (SSTP) möglich, welches von Microsoft entwickelt wurde.[74]

3.3.2 TLS

Das Transport Layer Security Protokoll (TLS) unterscheidet sich nur geringfügig von der dritten Version von SSL. Tatsächlich unterscheiden sich die zweite und die dritte Version von SSL deutlich mehr. Was genau TLS von SSL v3.0 unterscheidet, zeigt Tabelle 4.

Unterschiede in…	SSL v3.0	TLS v1.0
Protokollversion in Nachrichten	3.0	3.1
Typen von Benachrichtigungen	12	23
Nachrichtenauthentifizierung	Ad hoc	Standard
Schlüsselerzeugung	Ad hoc	PRF
Überprüfung der Zertifikate	komplex	Einfach
„Finished“-Nachricht	Ad hoc	PRF

Tabelle 4: Unterschiede zwischen TLS und SSL v3.0, Quelle: vgl. Thomas (2000), S.118.

Da, wie schon oben erwähnt, TLS eine Weiterentwicklung des SSL v3.0 ist, wurde auch die **Protokollversion** für die erste Version von TLS mit 3.1 festgelegt. Dies mag zwar am Beginn verwirrend klingen, wurde aber so festgelegt.

Einer der Bereiche, in denen TLS einen Vorteil gegenüber SSL bringt, ist die **Abarbeitung der Benachrichtigungen**. Im TLS-Protokoll wurden fast doppelt so viele Benachrichtigungsmöglichkeiten eingebaut wie in SSL.

Des Weiteren änderte sich der Algorithmus für **Nachrichtenauthentifizierung**. Während bei SSL die Kombination zwischen Schlüsselinformationen und Nutzdaten adhoc passiert, wird bei TLS der H-MAC[75]-Algorithmus angewandt. Dieser definierte Standard schützt Daten wie die Nachrichtenlänge, den Nachrichteninhalt, die TLS-Version und noch weitere.

Basierend auf diesem H-MAC-Algorithmus wird auch ein sogenannter „Pseudorandom Output“ erzeugt, welcher aus den zu verschlüsselnden Daten und einem Initialwert besteht und als Basis für die **Schlüsselerzeugung** herangezogen wird. Durch die Erstellung von nacheinander folgenden H-MAC-Werten ist es möglich, einen sicheren zufälligen Output zu generieren. Durch diesen „Pseudorandom Output“ kann dann im nächsten Schritt eine „Pseudorandom-Funktion“ (PRF) erstellt werden, aus welcher dann Schlüsseldaten erzeugt werden können.

[74] Vgl. Kohne/Ringleb/Yücel (2015), S.128.

[75] H-MAC…Hashed Message Authentication Code.

Während bei SSL die Funktion „CertificateVerify“ zur **Überprüfung der Zertifikate** komplexer von statten geht (2-stufiges Hash-Verfahren), erfolgt dies bei TLS durch das Austauschen der Handshake-Messages vor der Session.

Eine Abänderung für TLS wurde ebenso bei der **„Finished“-Nachricht** vorgenommen, welche das Ende der Verhandlungen und die Gültigkeit der vereinbarten Verschlüsselungsparameter kennzeichnet. TLS bezieht sich dabei wieder auf die Erstellung einer PRF.[76]

3.4 Vergleich von Protokollen

Um nun einen guten Überblick über die zuvor beschriebenen Protokolle zu erhalten, wird in folgender Tabelle ein Vergleich dargestellt. Dabei werden PPTP, L2TP, IPSec und SSL gegenübergestellt. Auf die Analyse der Protokolle L2F und TLS wird verzichtet, da L2F für den heutigen Einsatz nicht mehr relevant ist, und TLS auf Basis von SSL entwickelt wurde und daher die Unterscheidungsmerkmale nur gering sind (siehe Kapitel 3.3.2 TLS).

	PPTP	L2TP	IPSec	SSL
Layer	2	2	3	4-7
Standardisiert (RFC)	Nein	Ja	Ja	Ja
Einrichtung	Schnelle Einrichtung	Schnelle Einrichtung	Aufwändigere Einrichtung	Schnelle Einrichtung
Datenverschlüsselung	Ja	Nein	Ja	Ja
Schlüsselmanagement	Nein	Nein	Ja	Ja
IP-Tunneling	Ja	Ja	Ja	Ja (SSTP)
Fazit	Gute Kompatibilität, hohe Geschwindigkeit, wenig Sicherheit	In Kombination mit IPSec gut einsetzbar	Für die Vernetzung von Netzwerken gut geeignet, niedrigere Geschwindigkeit als PPTP, sichere Lösung	Für sichere Transaktionen gut geeignet, in Kombination mit IPSec noch mehr Einsatz-Szenarien denkbar

Tabelle 5: Vergleich der VPN-Protokolle, Quelle: vgl. Lipp (2001), S.90, vgl. Elektronik Kompendium (2016), Online-Quelle [26.Juli.2016], vgl. Remus (2015), Online-Quelle [26.Juli.2016].

[76] Vgl. Thomas (2000), S.118ff.

4 ANALYSE UND EMPFEHLUNG FÜR EIN SICHERES REMOTE-MONITORING BEI LOGICDATA

Nach der detaillierten Erläuterung der derzeit (mehr oder weniger) eingesetzten Protokolle für VPN-Verbindungen widmet sich dieses Kapitel nun den Sicherheitslücken, der aktuellen VPN-Situation bei LOGICDATA, sowie alternativer Protokolle, deren Einsatz für LOGICDATA am Ende des Kapitels diskutiert wird.

4.1 Mögliche Angriffsszenarien auf VPN-Verbindungen

Alle Chancen und Freiheiten, die der Einsatz des Internets mit sich gebracht hat, so viele Möglichkeiten gibt es auch, Benutzern des Internets zu schaden. Davon sind alle Einsatzbereiche betroffen – egal ob privat, geschäftlich oder gar Regierungen. Umso wichtiger ist es, sich dem Thema der Attacken auf VPN-Verbindungen zu widmen, als auch den Möglichkeiten, solche zu verhindern.[77]

Genau diese Möglichkeit bietet ein sogenanntes „**VPN Threat Model**“. Bei der Erstellung eines solchen Modells wird die Sicherheit des Netzwerkes optimiert, indem Schwachstellen aufgezeigt und Gegenmaßnahmen definiert werden. Am Ende dieses Prozesses sollte ersichtlich sein, welchem Bereich man die größte Aufmerksamkeit schenken sollte, um die Sicherheit des Systems zu gewährleisten. Den größten Mehrwert bringt das Modell natürlich, wenn dieser Prozess regelmäßig wiederholt wird, da sich Schnittstellen, Applikationen o.Ä. im Laufe der Zeit ändern und mögliche Schwachstellen darstellen können.[78]

Ein unter Sicherheitsexperten bekanntes Modell ist das „**STRIDE“-Modell**. Dessen Name setzt sich aus den Anfangsbuchstaben der darin enthaltenen möglichen Angriffe zusammen:

Angriffsbezeichnung	Erläuterung
Spoofing identity	Nutzen einer anderen Identität; z.B. Anmeldung mit illegal erhaltener Benutzerdaten
Tampering with data	Modifikation von Daten durch unbefugte Dritte; z.B. Man-in-the-middle-attack
Repudiation	Wenn eine böswillige oder unberechtigte Modifikation von Daten durchgeführt wurde, aber nicht mehr nachweisbar ist, durch wen das geschah
Information disclosure	Informationen gelangen an unberechtigte Dritte
Denial of service	Überlastung des Netzwerkes (genauere Erläuterung im nächsten Absatz)
Elevation of privilege	Unberechtigte Personen gelangen an Administrator-Rechte und richten Schäden damit an

Tabelle 6: Angriffsszenarien des STRIDE-Modells, Quelle: vgl. Henmi/Lucas/Singh/Cantrell (2006), S.320f.

[77] Vgl. Schudel/Smith (2008), o.S.

[78] Vgl. techtarget.com (2006), Online-Quelle [29.Juli.2016].

Als konkrete Angriffsszenarien für VPN-Netzwerke sind u.a. folgende bekannt:

Beim **IP-Spoofing** wird der IP-Header einer Nachricht so verändert, dass beim Empfänger eine andere als die wahre Quell-Adresse vorliegt. Dies wird durchgeführt, um die Nachricht auch über Firewalls versenden zu können. Die Nachricht würde bei Aufweisen einer anderen Quell-Adresse ausgefiltert werden. Ein Schutz gegen IP-Spoofing ist bereits in vielen Firewalls integriert.

DoS- (Denial of Service) Attacken werden von Hackern mit dem Ziel durchgeführt, das Netzwerk durch Überlastung zu stören, bis es sprichwörtlich zusammenbricht. Da bereits Software für DoS-Attacken auf Hackerwebseiten frei zum Download verfügbar sind, ist es sehr einfach einen solchen Angriff durchzuführen. Konkret wird die Überlastung durch eine hohe Netzwerkauslastung (Verschicken vieler Pakete) herbeigeführt, die die Netzwerkkomponenten (Server, Router o.Ä.) nicht mehr verarbeiten können.

Port-Scanning im Allgemeinen wird für die Auflistung aller „zuhörenden" Ports verwendet und führt beim alleinigen Einsatz noch zu keiner Störung des Rechners oder des Servers. Allerdings werden einem potentiellen Hacker damit Informationen zur Verfügung gestellt, auf welchem Weg er am besten ins Netzwerk kommen könnte. Vergleichbar ist dieses Szenario mit einem Autodieb, der die Türen der Autos überprüft und herausfinden möchte, welche offen sind. Sichtbar sind diese Abfragen auf der Firewall. Laufende „Ping"-Abfragen auf verschiedenen Ports von unterschiedlichen IP-Adressen machen darauf aufmerksam.[79]

Beim **Session Hijacking** wird die Kontrolle über die Verbindung nach der Authentifikation des Clients auf dem Server übernommen. Dabei sammelt der Hacker zunächst notwendige Informationen über die Session. Danach werden von ihm gefälschte Datenpakete eingeschleust.

Eine sehr häufige Attacke kann durch einen **Virus bzw. eine Malware** durchgeführt werden. Wenn der Client mit einem Virus infiziert wurde, ist es ein Leichtes für den Hacker an Zugangsdaten zu kommen. Sollte im Netzwerk keine Anti-Virus-Software aktiv sein, kann sich der Virus äußert schnell weiter ausbreiten und zusätzlichen Schaden anrichten.[80]

Bei einem Remote-Access-VPN ist es meistens der Fall, dass dem Benutzer durch „**Split Tunneling**" sowohl die öffentliche Verbindung ins Internet als auch die private Verbindung über VPN in das interne Netzwerk zur Verfügung steht. Wenn auf dem Client, der mit dem VPN verbunden ist, keine Sicherheitsmaßnahmen getroffen wurden, besteht die Möglichkeit einer Attacke auf das interne Netzwerk durch die öffentliche Verbindung. Aus diesem Grund ist es sinnvoll, das „Split Tunneling" abzuschalten.

Eine Hilfestellung für die Verhinderung solcher Attacken stellt z.B. „**VPN Separation**", also das Trennen von Adressierungsbereichen und Traffic von anderen VPN-Netzwerken, dar.

Durch die Architektur ist vorgegeben, dass nur PE-Router (Provider Edge) die VPN-Routen kennen müssen (im exklusiven IPv4-Adressbereich). In der Praxis wird die VPN-Verbindung zwischen dem CE-Router (Customer Edge) und einem Interface auf dem PE-Router hergestellt. Da dieses Interface eine

[79] Vgl. Henmi/Lucas/Singh/Cantrell (2006), S.88ff.

[80] Vgl. ClickSSL (2013), Online-Quelle [14.September.2016].

Adresse aus dem VPN-Adressbereich besitzt, ist eine Trennung zwischen dem „Core“ und dem VPN-Bereich möglich.

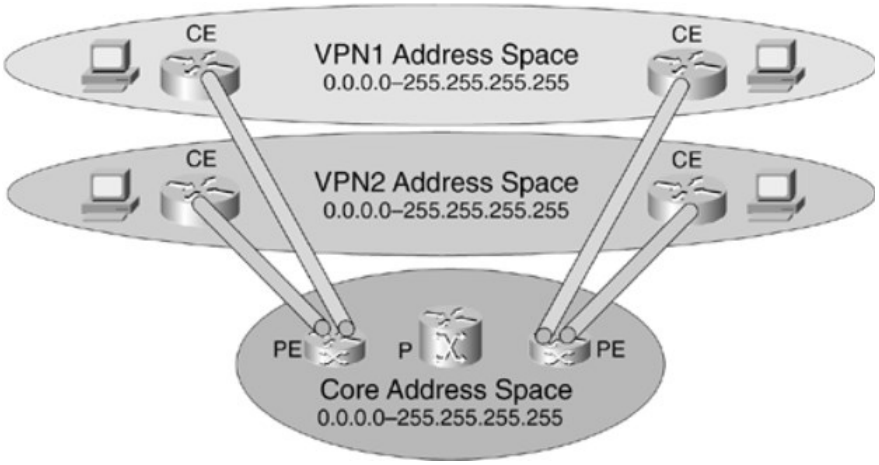


Abbildung 24: Address Seperation, Quelle: Behringer/Morrow (2005), Online-Quelle [29.Juli.2016].

Eine weitere Anforderung ist die Traffic-Trennung zwischen VPNs. Das heißt, dass Pakete nicht „vermischt“ werden und damit kein anderes VPN beeinflussen. Der Traffic wird in zwei Teile unterteilt:

- „Data Plane“ Traffic: Beinhaltet die Daten aus den einzelnen VPNs
- „Control Plane“ Traffic: Regelt den Ablauf im „Core“-Netzwerk

Wie in Abbildung 25 ersichtlich, wird der gesendete Traffic von PE zu PE gesendet. Dies geschieht in verkapselter Form und wird vom „Core“-Netzwerk nicht erkannt. Weitere Voraussetzungen für die Traffic-Trennung sind[81]:

- P-Router soll so ausgelegt werden, dass er von VPN nichts versteht und somit auch nicht für eine Vermischung des Traffics sorgt
- Das Interface am PE-Router muss logisch zum VPN gehören

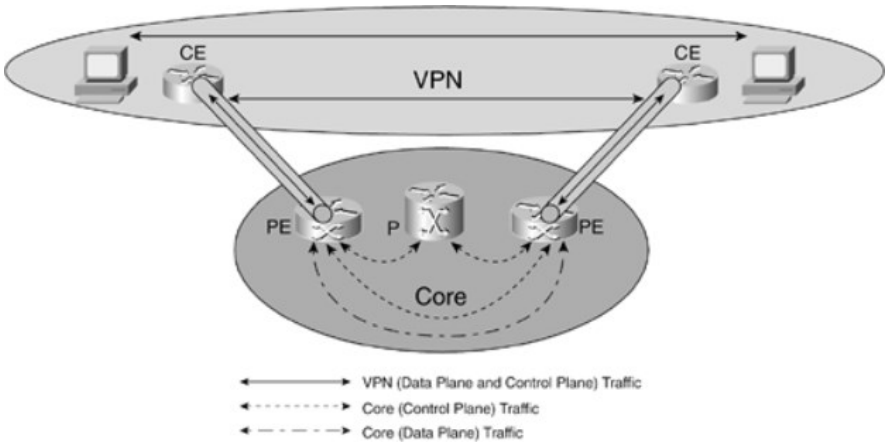


Abbildung 25: Traffic Seperation, Quelle: Behringer/Morrow (2005), Online-Quelle [29.Juli.2016].

[81] Vgl. Behringer/Morrow (2005), Online-Quelle [29.Juli.2016].

Generell werden für die Vermeidung oben genannter Attacken folgende Maßnahmen empfohlen[82]:

- Einsatz von Firewalls
- Für ein effektives Überwachen von Attacken wird ein IDS/IPS[83] empfohlen
- Installation einer Anti-Virus-Software auf allen Clients und Servern
- Systeme ohne Authentifizierungsmechanismus sollten keine Möglichkeit haben, sich über VPN mit dem internen Netzwerk zu verbinden
- Einschulung von Mitarbeitern (z.B. Administratoren, Remote-User, Support-Mitarbeiter) in Bezug auf IT-Sicherheit
- Definition von IT-Guidelines
- Strenge Vorgaben für die Passwortvergabe
- Einstiegspunkt des VPNs in einer DMZ[84], um das interne Netzwerk zu schützen
- „Split Tunneling“ unterbinden

4.2 Analyse der derzeitigen Situation bei LOGICDATA

Wie aus Abbildung 4: Netzwerkdiagramm zu Lohnfertigern, Quelle: Eigene Darstellung“ bereits hervorgeht, besteht zu den Servern, die bei den jeweiligen Lohnfertigern lokalisiert sind, eine Site-to-Site-Verbindung über einen IPSec-Tunnel. Als Technologie wird IPSec aus den folgenden Gründen eingesetzt:

- Hohes Maß an Sicherheit
- Stabilität der Verbindung

SSL als VPN-Protokoll wird zurzeit bei der Fa. LOGICDATA nur für die Verbindung von Mitarbeitern verwendet, die sich nicht in Deutschlandsberg befinden, wie z.B. für Home-Office-Mitarbeiter (Topologie: Remote-Access-VPN). Ein großer Vorteil ist hier die gute Funktionalität und Stabilität der Verbindung weltweit, sowie die einfache Einrichtung auf den Geräten.

4.3 Kriterien für die Auswahl einer VPN-Lösung

Bei der Auswahl der VPN-Lösung, speziell zu den Lohnfertigern, waren vor allem die ausreichenden Sicherheitsmaßnahmen von Bedeutung. Der bestehende Tunnel, der die Daten sicher transportiert, ist ein essentielles Kriterium. Außerdem muss auch auf die Verschlüsselung und den Algorithmus ein besonderes Augenmerk gelegt werden. Am besten wäre die Verwendung eines veröffentlichten

[82] Vgl. The Government of the Hong Kong Special Administrative Region (2008), Online-Quelle [14.September.2016], S.20f, vgl. ClickSSL (2013), Online-Quelle [14.September.2016], Frahim/Huang (2008), Online-Quelle [14.September.2016].

[83] IDS/IPS…Intrusion Detection System/Intrustion Prevention System.

[84] DMZ…Demilitarised Zone.

Algorithmus, da an diesem bereits schon einige Angriffstests stattgefunden haben. Sollte er dann immer noch sicher sein, ist davon auszugehen, dass die Daten noch einige Zeit durch einen Angriff nicht entschlüsselt werden können.

Für den täglichen Betrieb ist es unbedingt notwendig, dass die Verbindung zu den Servern stabil und zuverlässig besteht. Ausfälle können in dieser Beziehung starke monetäre Auswirkungen haben. Zusätzlich ist auch zu bedenken, dass täglich ungefähr 500MB übermittelt werden müssen. Die Verbindung muss daher auch auf die Übermittlung größerer Datenmengen ausgelegt sein. Für den Fall der Erweiterung um einen Lohnfertiger ist initial eine einfache Einrichtung der Verbindung von Vorteil.

4.4 Alternative Protokolle

Wie in Kapitel 3 „VPN-Protokolle“ bereits erläutert, existieren eine Reihe von Möglichkeiten, durch bewährte Protokolle eine stabile und sichere VPN-Verbindung aufzubauen. Im folgenden Kapitel soll der Fokus allerdings auf Alternativen zur „herkömmlichen“ VPN-Verbindung liegen und analysiert werden, ob auch andere Protokolle für einen ähnlichen Einsatz geeignet sind.

4.4.1 MQTT

Das MQTT- (Message Queue Telemetry Transport) Protokoll wurde 1999 von Dr. Andy Stanford-Clark (IBM) und Arlen Nipper (Arcom) erfunden. Seit 2013 ist es standardisiert und als aktuelle Version MQTT V3.1. verfügbar. Der primäre Einsatz des Protokolls liegt zurzeit im Vernetzen von Geräten oder Maschinen im „Internet of Things“-Bereich. Ein besonderes Augenmerk wurde bei der Entwicklung einerseits auf eine minimale Auslastung der Netzwerk-Bandbreite und andererseits auf Verlässlichkeit (gewisser Grad an Garantie, dass die gesendeten Pakete auch tatsächlich ankommen) gelegt. Es wurde versucht, das Protokoll so einfach wie möglich und gleichzeitig so verlässlich wie möglich zu gestalten. Dies bringt natürlich Einschränkungen in Bezug auf die Sicherheit.

Mithilfe des Protokolls können zwar ein Benutzername und ein Passwort übermittelt werden, eine Verschlüsselung durch MQTT ist jedoch nicht möglich. Hierfür kann zusätzlich SSL bzw. TLS eingesetzt werden, das das Netzwerk allerdings wesentlich höher auslastet. Eine weitere Möglichkeit für die Übertragung verschlüsselter Daten wäre der Einsatz von Applikationen, die diese Verschlüsselungsaufgabe übernehmen. Diese Einschränkungen müssen in Kauf genommen werden, da das Protokoll in seiner Einfachheit erhalten bleiben soll.[85]

Das MQTT-Protokoll ist ein sogenanntes „publish/subscribe“-Protokoll, welches in einer Client/Server-Umgebung eingesetzt wird. Dabei stellen bei einem Einsatz im IoT-Bereich Sensoren die Clients dar. Der Server wird in diesem Kontext auch „Broker“ genannt. Clients und Server sind über TCP miteinander verbunden (UDP wird aufgrund der unzureichenden Zuverlässigkeit nicht für MQTT eingesetzt). Jede Nachricht stellt einen separaten Datensatz dar, der für den Broker unverständlich ist.

Der Ablauf der Nachrichtenübertragung ist in Abbildung 26 dargestellt. Jede Nachricht wird für eine bestimmte Adresse veröffentlicht. Diese Adresse wird „Topic“ genannt. Jeder Client kann mehrere Topics

[85] Vgl. MQTT.org (2016), Online-Quelle [30.Juli.2016].

„abonnieren“. So werden die Daten bei Veröffentlichung einer Nachricht von allen Clients erhalten, die sich dafür angemeldet haben.

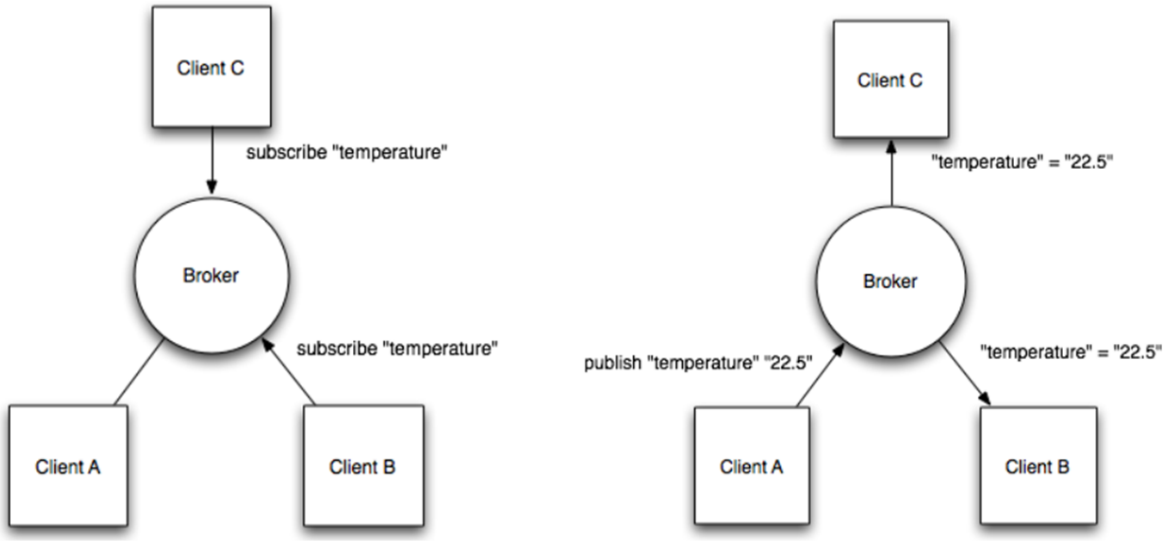


Abbildung 26: MQTT-Architektur, Quelle: Jaffey (2014), Online-Quelle [30.Juli.2016].

Ein gutes Beispiel hierfür zeigt Abbildung 26. Am Beginn der Übertragung muss die Festlegung der Topics durch die Clients erfolgen. (Clients A und B abonnieren das Topic „temperature“.) Danach veröffentlicht Client A einen konkreten Temperaturwert. Der Broker leitet den Temperaturwert nun an alle Clients weiter, die sich dafür interessieren.

Bei einem Stau der Nachrichten im Broker werden immer nur die letzten Nachrichten für einen jeweiligen Client behalten. Dies kann passieren, wenn der Client beim Broker eine Anfrage für das „Nicht-Senden“ einer Nachricht stellt.[86]

Eine Weiterentwicklung des MQTT-Protokolls wurde erstmals 2008 präsentiert – das MQTT-SN[87]-Protokoll. Dieses wurde speziell für die verbindungslose Kommunikation konzipiert und so gestaltet, dass es plattform-unabhängig funktioniert. D.h. jedes Netzwerk, welches eine bidirektionale Übertragung zwischen zwei Endgeräten zur Verfügung stellt, ist für den Einsatz von MQTT-SN geeignet. Die Architektur unterscheidet sich ein wenig von der MQTT-Architektur (siehe Abbildung 27). Drei verschiedene Komponenten werden für eine erfolgreiche Übertragung benötigt: „Clients“, „Fowarders“ und „Gateways“. Clients können sich entweder direkt mit dem „Broker“ über das Gateway verbinden, oder auch zuerst über einen Forwarder (wenn sich das Gateway nicht im selben Netzwerk befindet). Das Gateway kann sowohl im Broker integriert sein, als auch als „stand-alone“-Variante ausgeführt sein (hier fungiert das MQTT-Protokoll dazwischen als Übersetzer zwischen MQTT und MQTT-SN). Der „Publish/Subscribe“-Mechanismus funktioniert beim MQTT-SN-Protokoll nach dem gleichen Prinzip wie bei MQTT.[88]

[86] Vgl. Jaffey (2014), Online-Quelle [30.Juli.2016].

[87] MQTT-SN…MQTT for Sensor Networks.

[88] Vgl. Stanford-Clark/Truong (2013), Online-Quelle [14.September.2016], S. 3ff.

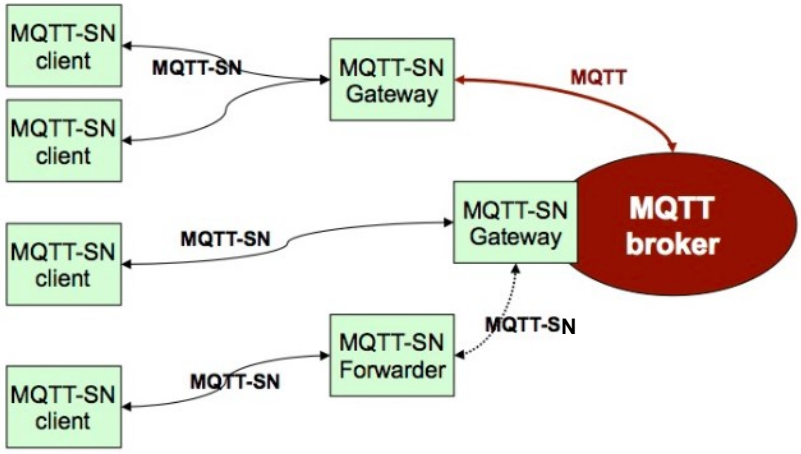


Abbildung 27: MQTT-SN-Architektur, Quelle: vgl. Stanford-Clark/Truong (2013), Online-Quelle [14.September.2016], S.6.

4.4.2 CoAP

Ein weiteres Protokoll, welches hauptsächlich für IoT-Netzwerke oder M2M[89]-Kommunikation eingesetzt wird, ist das Constrained Application Protocol (CoAP). Dieses spezielle Web-Transferprotokoll wurde vor allem für Netzwerke entwickelt, die besonders auf niedrigen Energieverbrauch Wert legen müssen. Auf Basis des Architekturmodells für Web-Services (REST[90]), welches Ressourcen von Servern über Applikationsprozesse kontrolliert und über URI[91] identifiziert, wurde CoAP entwickelt.[92]

Wie aus Abbildung 28 hervorgeht, ist das Protokoll zwischen der Transport- und der Applikationsschicht angesiedelt. Im Gegensatz zu HTTP setzt das CoAP auf das UDP auf, welches einen wesentlich kleineren Overhead hat.

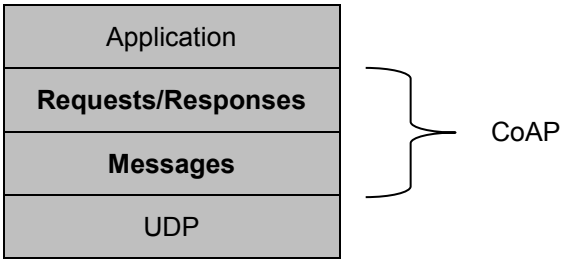


Abbildung 28: Aufbau des CoAP, Quelle: In Anlehung an: Request for Comments: 7252 (2014), S.9.

Die untere Schicht des CoAP bildet die „Messages“-Schicht, die für die Kommunikation mit UDP verantwortlich ist. Darüber befindet sich die „Requests/Responses“-Schicht, die die „Request“- und „Response“-Nachrichten verarbeitet und so die Kommunikation steuert.[93]

[89] M2M…Mashine to Mashine.

[90] REST…Representational State Transfer.

[91] URI…Uniform Resource Identifiers.

[92] Vgl. Chen (2014), Online-Quelle [1.August.2016].

[93] Vgl. Request for Comments: 7252 (2014), S.4.

Die „Message“-Schicht stellt vier Nachrichtentypen zur Verfügung: CON (confirmable), NON (non-confirmable), ACK (Acknowledgement) und RST (Reset). Bei den Verbindungen zwischen Server und Client wird zwischen einer verlässlichen und unverlässlichen Nachrichtenübertragung unterschieden.

Bei einer **verlässlichen Nachrichtenübertragung** wird vom Client am Beginn ein CON mit der Nachrichten-ID gesendet. Danach wird auf die Antwort des Servers gewartet, der bei erfolgreichem Verbindungsaufbau mit ACK und derselben Nachrichten-ID antwortet. Hat der Aufbau fehlgeschlagen, wird die Nachrichten-ID durch ein RST ersetzt.

Im Gegensatz zur **unverlässlichen Nachrichtenübertragung**, wo vom Client ein NON mit der Nachrichten-ID gesendet wird. Hier wird vom Server nur ein RST gesendet, wenn der Aufbau nicht erfolgreich stattgefunden hat. Ansonsten wird bei erfolgreichem Verbindungsaufbau keine Antwort vom Server erwartet. Diese Art der Nachrichtenübertragung verwendet man z.B. bei periodischen Abfragen.

Auf der „Requests/Responses“-Schicht findet die tatsächliche Kommunikation statt. Hier sind ebenso wieder drei Kommunikationsszenarien möglich.

Bei der sogenannten „Piggy-backed“-Kommunikation sendet der Client eine CON-Nachricht. Darauf folgt umgehend eine ACK-Antwort vom Server, die die Meldung enthält, ob die Daten erfolgreich abgefragt werden konnten (siehe Abbildung 29).[94]

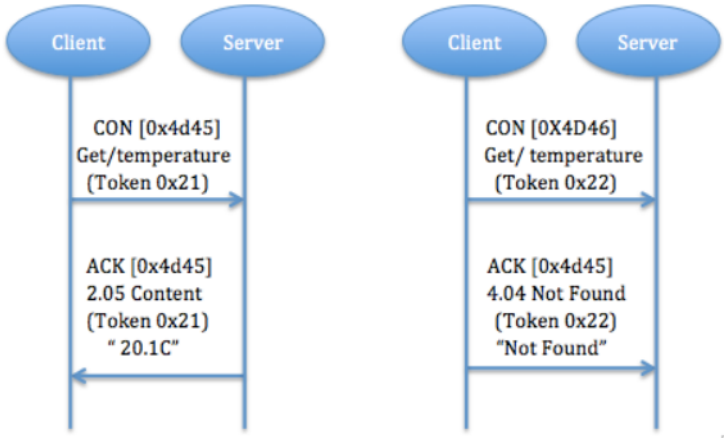


Abbildung 29: "Piggy-backed"-Kommunikation, Quelle: Chen (2014), Online-Quelle [1.August.2016].

Wird eine „Separate response“-Kommunikation durchgeführt, wird nach dem CON vom Client sofort eine leere ACK-Antwort gesendet um den Empfang des CON zu bestätigen. Dabei soll eine weitere CON-Nachricht an den Server verhindert werden. Wenn der Server die angefragten Daten bestätigen kann, wird ein erneutes CON (diesmal vom Server an den Client) mit den angefragten Daten gesendet. Zur Empfangsbestätigung der CON-Nachricht erhält der Server am Ende ein erneutes ACK vom Client (siehe Abbildung 30).[95]

[94] Vgl. Chen (2014), Online-Quelle [1.August.2016], vgl. Request for Comments: 7252 (2014), S.10ff.

[95] Vgl. Chen (2014), Online-Quelle [1.August.2016], vgl. Request for Comments: 7252 (2014), S.12ff.

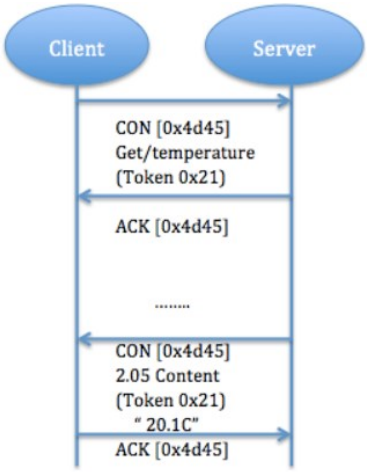


Abbildung 30: "Seperate response"-Kommunikation, Quelle: Chen (2014), Online-Quelle [1.August.2016].

Das letzte Kommunikationsszenario ist das „Non confirmable request and response“ (siehe Abbildung 31). Hier wird im Gegensatz zur „Piggy-backed“-Kommunikation vom Client eine NON-Nachricht gesendet, die signalisiert, dass vom Server keine Empfangsbestätigung erwartet wird. Die Antwort des Servers wird ebenfalls eine NON-Nachricht sein, welche die angefragten Werte beinhaltet (wenn erfolgreich).[96]

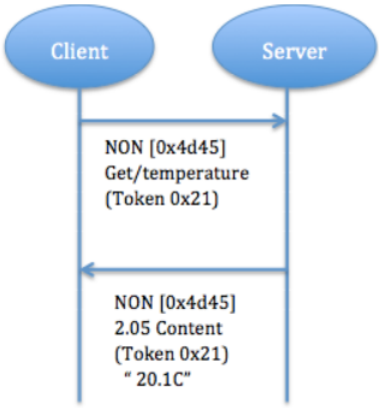


Abbildung 31: "Non confirmable request and response"-Kommunikation, Quelle: Chen (2014), Online-Quelle [1.August.2016].

Um die Sicherheit von CoAP-Verbindungen sicherstellen zu können, wird ein eigenes Protokoll eingesetzt – das „Datagram Transport Layer Security“-Protokoll (DTLS). Es sorgt für die Einhaltung von Integrität, Authentizität und Vertraulichkeit. Während HTTP TLS (über TCP) für eine sichere Verbindung nutzt, verwendet CoAP DTLS (über UDP). DTLS basiert auf TLS, nur mit der Zusatzfunktion, mit dem unverlässlichen Verhalten von UDP umzugehen. DTLS stellt eine Ende-zu-Ende-Verschlüsselung zur Verfügung, welche die Daten vor Angriffen gut schützen kann. [97]

[96] Vgl. Chen (2014), Online-Quelle [1.August.2016], vgl. Request for Comments: 7252 (2014), S.13.

[97] Vgl. Chen (2014), Online-Quelle [1.August.2016], vgl. Request for Comments: 7252 (2014), S.67f.

4.4.3 TINA

Das „Telecommunications Information Networking Architecture“- (TINA) Protokoll ist ein von der Firma „Barracuda“ entwickeltes VPN-Protokoll, welches auf Basis von IPSec aufgebaut wurde. Im Speziellen sorgt es für eine höhere Zuverlässigkeit und Sicherheit der VPN-Verbindung.[98]

Besondere Features von TINA (gegenüber IPSec) sind folgende[99]:

- Auswählen verschiedener Transport-Modi
    - Transport-Modi: TCP, UDP, TCP & UDP
    - Diese Möglichkeit kann, je nach lokaler Gegebenheit, die Performance und die Stabilität der Verbindung erhöhen
- Traffic Intelligence
    - TINA unterstützt die Auswahl alternativer Kommunikationswege über einen verschlüsselten Tunnel
- Traffic Shaping
    - Optimale Nutzung der Bandbreite im VPN-Tunnel
- Data Compression

Bei der Nutzung des TINA-Protokolls ist allerdings auch zu erwähnen, dass nur ein kleiner Kreis an unterstützten Geräten von ausgewählten Herstellern für ein VPN-Netzwerk einsetzbar ist. Darauf muss beim Einsatz dieses Protokolls geachtet werden.

4.5 Empfehlung einer Alternative

Die zuvor beschriebenen Alternativen eröffnen eine Reihe neuer Möglichkeiten. Der Einsatz von IoT-Protokollen (MQTT und CoAP) im Speziellen, ist in sehr vielen Bereichen möglich. In den folgenden Absätzen wird nun Bezug auf die Lohnfertigersituation bei LOGICDATA genommen und der Einsatz der alternativen Protokolle diskutiert.

Das MQTT-Protokoll ist im Hinblick auf die Topologie (Hub-and-Spoke) durchaus für einen Einsatz bei LOGICDATA denkbar (siehe Abbildung 4 „Netzwerkdiagramm zu Lohnfertigern, Quelle: Eigene Darstellung“). Eine einfache Erweiterung eines Lohnfertigers ist daher möglich. Des Weiteren liegt die Übertragungsgeschwindigkeit im Sekundenbereich, die in diesem Fall allerdings nicht störend ist, da die Daten ohnehin nur einmal pro Tag (über Nacht) übertragen werden. Da das MQTT-Protokoll TCP nutzt, kann einerseits eine verlässliche Verbindung aufgebaut werden, andererseits werden auch Ressourcen (wie die Bandbreite) stärker beansprucht. Dies stellt für LOGICDATA kein Problem dar, da keine Ressourcenknappheit besteht. Ein Einsatz von MQTT-SN wird daher auch nicht angedacht. Da das MQTT-Protokoll an sich die Daten unverschlüsselt überträgt, muss die Sicherheit bei einem Einsatz bei

[98] Vgl. Barracuda Networks (2015), Online-Quelle [12.August.2016].

[99] Vgl. MILS Electronic (2010), Online-Quelle [12.August.2016], S.3f.

LOGICDATA unbedingt durch SSL bzw. TLS gewährleistet werden. Die maximale Übertragungsmenge von 256MB/Paket würde beim Gebrauch von MQTT bei LOGICDATA eine Aufteilung der Daten erfordern.[100]

Das CoAP-Protokoll eignet sich allein durch die vorgegebene Topologie (Point-to-Point) eher nicht für die Anwendung bei LOGICDATA. Weitere Nachteile sind die geringe zu übertragende Datenmenge (1024 Bytes/Paket) sowie der Einsatz von UDP. Die Sicherheit könnte zwar durch DTLS abgedeckt werden, insgesamt weist das Protokoll aber eher geringes Anwendungspotential für LOGICDATA auf.[101]

Das TINA-Protokoll weist zwar höhere Sicherheitsmaßnahmen als IPSec auf, erfordert aber bei einer Umstellung einen Austausch aller betroffenen Netzwerkkomponenten durch TINA-fähige Geräte (da TINA nur auf einer eingeschränkten Gruppe von Netzwerkkomponenten unterstützt wird). Sollte es in Zukunft zu einem Austausch der Geräte kommen, sollte TINA als alternatives VPN-Protokoll berücksichtigt werden. Für eine sofortige Umstellung wären die Kosten zu hoch.

Zusammenfassend ist festzuhalten, dass IoT-Protokolle eigentlich für einen anderen Anwendungsbereich ausgelegt sind – nämlich für die Übertragung vieler und kleiner Datenmengen in kurzer Zeit. Dies spiegelt genau das Gegenteil der Lohnfertiger-Situation bei LOGICDATA wider (Übertragung einer großen Datenmenge einmal pro Tag). Ein Einsatz des MQTT-Protokolls wäre zwar theoretisch möglich, allerdings aus der Sicht des Autors nicht optimal. Die Anwendung des TINA-Protokolls statt der derzeitigen IPSec-VPN-Lösung wäre, aus den zuvor beschriebenen Gründen, in Zukunft denkbar.

[100] Vgl. Schneider (2013), Online-Quelle [22.August.2016], vgl. Gupta (2014), Online-Quelle [22.August.2016], vgl. Stansberry (2015), Online-Quelle [22.August.2016].

[101] Vgl. Request for Comments: 7252 (2014), S.25, Stansberry (2015), Online-Quelle [22.August.2016].

5 MONITORING IN DER THEORIE

Nach Betrachtung der (VPN-)Verbindung zu den Lohnfertigern, die ein sicheres Remote-Monitoring möglich macht, widmet sich dieses Kapitel nun den Möglichkeiten, wie der Zustand von Netzwerkkomponenten (konkret bei LOGICDATA: von Servern) überprüft werden kann. Vor der tatsächlichen Installation der Monitoring-Tools in der Testumgebung wird zuvor noch auf die Mechanismen eingegangen, die für das Monitoring benötigt werden.

5.1 SNMP

Das Simple Network Management Protocol (SNMP) wurde 1988 entwickelt, um SNMP-unterstützende Geräte (z.B. Router, Firewalls, etc.) mit wenigen Befehlen verwalten zu können.

Wie aus Abbildung 32 hervorgeht, geschieht die Kommunikation zwischen einer sogenannten „Network Management Station“ (NMS), welche z.B. ein Server sein kann, und einem Agent, welcher eine Software auf dem zu überwachenden Gerät darstellt. Dabei werden vom Agent „Traps“ gesendet um die NMS darüber zu informieren, dass sich etwas verändert hat (z.B. ein bestimmter Wert). Diese Traps werden asynchron vom Agent gesendet, das heißt, dass die NMS diese zuvor nicht angefragt hat. Auf diese von der NMS empfangenen Traps reagiert diese mit Queries. Diese sollen beim Agent bestimmte Informationen abfragen, die er dann wieder zurück an die NMS sendet.

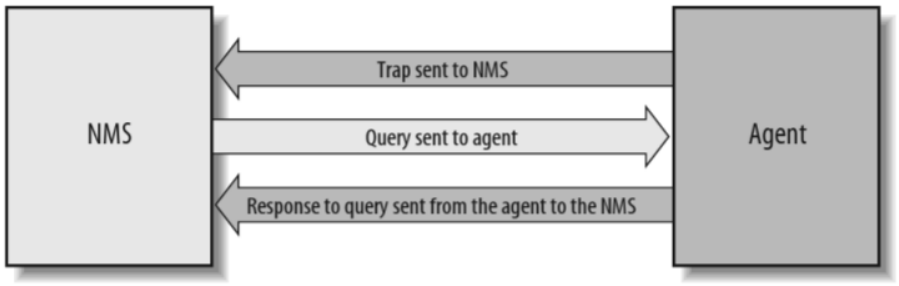


Abbildung 32: Funktionsweise SNMP, Quelle: Mauro/Schmidt (2005), S.4.

SNMP verwendet UDP als Transportprotokoll, welches zwar unzuverlässig ist, da es keine Informationen über Paketverluste gibt, allerdings verbindungslos funktioniert. Die Überprüfung über einen Paketverlust wird von der SNMP-Applikation durchgeführt, welche über ein Timeout realisiert wird.

Um Netzwerkkomponenten über SNMP steuern zu können, bedarf es der Management Information Base (MIB), die eine Reihe von zu steuernden Netzwerkobjekten enthält (z.B. Interfaces auf einem Router). Während die „Structure of Management Information“ (SMI) einen Weg für das Definieren von zu überwachenden und zu steuernden Objekten bereitstellt, ist die MIB für das Definieren an sich verantwortlich. D.h. darin enthalten sind ein Name des Objektes, sowie dessen Bedeutung. Beim Kauf eines Routers von Cisco z.B. ist die MIB schon erstellt und steht für den Benutzer zur Verfügung. Allerdings kann auch eine eigene MIB erstellt werden, sofern dies gewünscht ist.[102]

[102] Vgl. Mauro/Schmidt (2005), S.4ff.

Mithilfe des Nachrichtenformats PDU (Protocol Data Unit) werden Informationen zwischen Manager und Agent ausgetauscht. Dafür werden folgende Kommandos verwendet:

Kommando	Verwendung in Version…	Erläuterung
Get	Alle	Vom NMS werden Informationen beim Agent abgefragt
Getnext	Alle	Muss für jeweils ein MIB-Objekt ausgeführt werden, um Werte dieses Objektes abzufragen
Getbulk	V2 & V3	Abfrage einer großen Datenmenge aus der MIB-Tabelle
Set	Alle	Modifizierung eines Wertes am Agent
Getresponse	Alle	Antwort vom Agent auf „Get“
Trap	Alle	Vom Agent gesendete Nachrichten (eventgesteuert)
Inform	V2 & V3	Informationsmechanismus für bestätigte Nachrichten
Report	V2 & V3	Um Probleme bei der Nachrichtenübertragung zu reporten

Tabelle 7: SNMP-Kommandos, Quelle: vgl. Mauro/Schmidt (2005), S.37ff, vgl. Zoho Corp. (2016), Online-Quelle [8.August.2016].

SNMP wurde als Standard von der IETF (Internet Engineering Task Force) veröffentlicht. Insgesamt wurden bis heute drei Versionen veröffentlicht.[103]

5.1.1 SNMPv1 & SNMPv2

SNMP Version 1 ist die ursprüngliche Version des Protokolls und ist im RFC 1157 definiert. Um Zugriff auf ein Gerät über SNMPv1 zu bekommen, muss lediglich ein Passwort angegeben werden, mit welchem man Informationen vom Gerät abfragen kann. Obwohl es schon Weiterentwicklungen von SNMP durch die Versionen 2 und 3 gab, ist die erste Version noch immer die auf den Netzwerkkomponenten am häufigsten von den Herstellern unterstützte.[104]

In den Versionen 1 & 2 von SNMP werden sogenannte „Communities“ verwendet, um die Verbindungsart zwischen Manager und Agent festzuhalten. Der Community-Name ist mit einem Passwort zu vergleichen, mit dem man dann auf einem Rechner, Server oder anderen Netzwerkkomponenten Zugriff erlangt. Es wird zwischen drei Communitiy-Arten unterschieden[105]:

- „Read-Only“: Daten können gelesen aber nicht modifiziert werden; Standardeinstellung: „public“
- „Read-Write“: Daten können gelesen und modifiziert werden; Standardeinstellung: „private“
- „Trap“: Erlaubt das Erhalten von Traps (asynchrone Benachrichtigungen) vom Agent

[103] Vgl. Request for Comments: 2570 (1999), S.1.

[104] Vgl. Mauro/Schmidt (2005), S.2ff.

[105] Vgl. Mauro/Schmidt (2005), S.21f.

5.1.2 SNMPv3

Die dritte Version von SNMP wurde auf Basis von SNMPv2 entwickelt. Neben den Features von Version 2, beinhaltet v3 insbesondere neue Sicherheitsfunktionen. Zwei Neuerungen sind[106]:

- USM (User-based Security Module): beinhaltet eine Liste von Benutzern und deren Attributen
- VACM (Version-based Access Control Module): steuert den Zugriff durch Benutzer auf Netzwerkkomponenten

5.2 Bandbreitenmessung

Für die Bandbreitenmessung kommen speziell zwei Technologien in Frage: NetFlow und SFlow. Beide benötigen einen Kollektor, der die Daten entsprechend anzeigt. Die Auswahl des Kollektors wird in Kapitel 6 „Auswahl einer Monitoring-Software“ tiefergehend beschrieben. In den folgenden Absätzen widmet sich der Autor der näheren Erläuterung von NetFlow und SFlow.

5.2.1 NetFlow

Die von der Firma Cisco entwickelte Technologie macht es für IT-Administratoren möglich, Informationen aus ihrem Netzwerk zu erhalten. Netzwerkkomponenten (wie z.B. Router, Switches oder Firewalls) sammeln diese „Flow“-Daten und exportieren diese zum sogenannten Kollektor. Dieser ist dann für die Darstellung der Daten verantwortlich. Ein Flow ist definiert als „*unidirectional sequence of packets with some common properties that pass through a network device*“[107] und ist granular aufgebaut (beinhaltet z.B. Informationen über die Quell- und Zieladresse, Portnummern, Interfaces, etc.). Das eröffnet die Möglichkeit einer guten Filterung durch den Kollektor. Bisher wurden drei Versionen von NetFlow veröffentlicht: v3, v5 und v9.[108]

Als Transportprotokoll wird UDP verwendet, v9 ist allerdings protokollunabhängig entwickelt worden. Außerdem ist ein Export an mehrere Kollektoren mithilfe von unterschiedlichen Transportprotokollen möglich. Bei einer überlastungsanfälligen Verbindung sollte beim Einsatz von NetFlow darauf geachtet werden, dass die stoßartigen Exporte die Verbindung nicht einbrechen lassen. Eine speziell dafür vorgesehene Verbindung ist daher zu empfehlen.[109]

Ein exportiertes Paket setzt sich neben dem immer vorgesehenem „Packet Header“ aus sogenannten „Flow Sets“ zusammen. Diese werden, wie aus Abbildung 33 ersichtlich, in drei Typen unterteilt[110]:

---

[106] Vgl. Mauro/Schmidt (2005), S.73ff.

[107] Request for Comments: 3954 (2004), S.1.

[108] Vgl. Request for Comments: 3954 (2004), S.1f.

[109] Vgl. Request for Comments: 3954 (2004), S.7.

[110] Vgl. Request for Comments: 3954 (2004), S.11ff.

- „Template Flow Set“: Durch Templates kann ein hoher Grad an Flexibilität erreicht werden, da der Kollektor die Flows verarbeiten kann, ohne notwendigerweise die Interpretation der Daten zu kennen. Dies geben die Templates vor.
- „Data Flow Set“: In diesem Flow Set befinden sich die tatsächlichen Daten und können mit dem zugehörigen Template vom Kollektor interpretiert werden.
- „Options Template Flow Set“: Stellt zusätzliche Informationen zur Verfügung (z.B. Abtastrate, Interface-Informationen ,etc.).

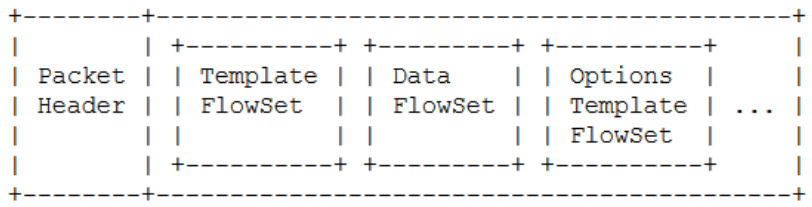


Abbildung 33: Paketaufbau NetFlow, Quelle: Request for Comments: 3954 (2004), S.8.

Die folgende Tabelle zeigt einen Auszug der Daten, die über NetFlow-Pakete zum Kollektor weitergeleitet und dort verarbeitet werden können:

Bezeichnung	Erläuterung
IN_BYTES	Eingehende Bytes, die zu einem Flow gehören
OUT_BYTES	Ausgehende Bytes, die zu einem Flow gehören
IPV4_SRC_ADDR	IPv4 Quell-Adresse
IPV4_DST_ADDR	IPv4 Ziel-Adresse
FLOWS	Anzahl der Flows, die aggregiert wurden
SAMPLING_INTERVAL	Abtastrate; wenn Wert z.B. 100, würde 1 Paket aus 100 Paketen abgefragt werden
SRC_MAC bzw. DST_MAC	Quell- bzw. Ziel-MAC-Adresse
IP_PROTOCOL_VERSION	IPv4 (Wert = 4) oder IPv6 (Wert = 6)

Tabelle 8: Daten in einem NetFlow-Paket, Quelle: vgl. Request for Comments: 3954 (2004), S.19ff.

Am Beginn der Entwicklungen von NetFlow wurde davon ausgegangen, dass sich der Exporter (z.B. Router) und der Kollektor in demselben privaten Netzwerk befinden. Da auch ein Transport über das öffentliche Netzwerk (Internet) möglich ist, muss bei Verwendung betont werden, dass sich daraus Sicherheitsrisiken ergeben.

Durch einen Angriff könnten exportierte Pakete einbehalten, modifiziert oder eingeschoben werden. NetFlow stellt keine Maßnahmen für die Sicherstellung von Integrität, Authentizität und Vertraulichkeit zur

Verfügung, da sich dies auf die Effizienz des Protokolls auswirken würde. Das IPFX[111]-Protokoll, welches auf NetFlow V9 basiert, deckt diese Sicherheitsaspekte ab.[112]

5.2.2 SFlow

Sflow ähnelt der Netflow-Technologie, allerdings wurde diese für eine Vielzahl von Geräten (und Herstellern) entwickelt. Ein SFlow-Monitoring-System besteht (ebenso wie bei NetFlow) aus einem Agent (integriert in z.B. einen Switch oder Router), und einem SFlow-Kollektor. Für die Verbindung zwischen diesen beiden wird SNMP eingesetzt.[113]

Der Prozess beim Exportieren der Daten aus dem Agent zum Kollektor ist der folgende:

1. 1 Paket aus N Paketen wird vom Switch oder Router abgefragt.
2. Der Switch oder Router gibt Informationen über den Header des abgefragten Pakets, Input- und Output-Interface und zusätzliche Parameter an den SFlow-Agent weiter. Dieser startet dann mit dem Aufbau des SFlow-Datenpaketes.
3. Das SFlow-Datenpaket wird an den Kollektor gesendet und dort analysiert.

Dieser Vorgang kann natürlich für beliebig viele Netzwerkkomponenten durchgeführt werden, dessen Daten von ein und demselben Kollektor analysiert werden.[114]

Das SFlow-Datenpaket enthält die in Abbildung 34 ersichtlichen Informationen:

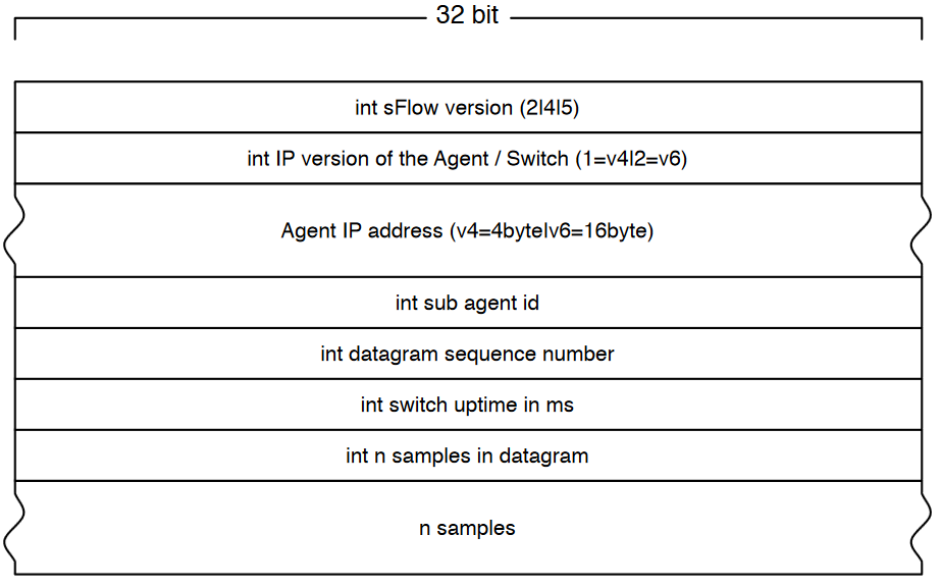


Abbildung 34: Aufbau eines SFlow-Datenpaketes, Quelle: Jasinska (2004), Online-Quelle [11.August.2016].

[111] IPFX… IP Flow Information eXport.

[112] Vgl. Request for Comments: 3954 (2004), S.26.

[113] Vgl. Phaal/Lavine (2004), Online-Quelle [11.August.2016], S.5f.

[114] Vgl. inMon Corp. (2007), Online-Quelle [11.August.2016].

SFlow stellt keine Sicherheitsmechanismen zur Verfügung. Daher sollte besonders auf folgende Aspekte Augenmerk gelegt werden, um Angriffe möglichst zu verhindern[115]:

- Richtige Konfiguration: Der Agent sollte mit der SFlow-MIB konfiguriert werden, da SNMP bereits ausreichende Sicherheit bietet.
- Transport: Da die Datenpakete vom Agent zum Kollektor unverschlüsselt übertragen werden, kann es zu Manipulationen (z.B. durch Spoofing) dieser kommen. Der Einsatz eines VPN-Tunnels ist daher ratsam.
- Vertraulichkeit: Die Traffic-Informationen aus den SFlow-Datenpaketen geben sehr tiefe Einblicke in ein Netzwerk. Der Zugang zum Kollektor sollte nur vertrauenswürdigen Stellen ermöglicht werden.

[115] Vgl. Phaal/Lavine (2004), Online-Quelle [11.August.2016], S.44f.

6 AUSWAHL EINER MONITORING-SOFTWARE

Nach der ausführlichen Erläuterung der Theorie, widmet sich dieses Kapitel dem Start des praktischen Teils der Arbeit. Die Ziele dafür wurden folgendermaßen definiert:

- Mit welcher Technologie kann ein Live-Monitoring realisiert werden?
- Welche Software ist für die Anforderungen geeignet?

Beide Fragen werden nun in den weiteren Kapiteln beantwortet, wobei der Fokus vorerst auf dem Vergleich der getesteten Lösung liegt. Die Auswahl einer Software wird in Kapitel 7 „Einführung einer ausgewählten Monitoring-Software“ detailliert beschrieben.

6.1 Testaufbau

Am Beginn der praktischen Aufgabenstellung musste eine geeignete Testumgebung (siehe Abbildung 35) aufgebaut werden, um ein komfortables Experimentieren zu erreichen. Dabei wurden natürlich die reellen Bedingungen nachempfunden. Es wurden zwei verschiedene Netzwerke eingerichtet (Wien, Netz: 10.0.0.1/24 & New York, Netz: 10.0.1.1/24), die ein Lohnfertiger-Netzwerk und das LOGICDATA-Netzwerk darstellen sollen. Verbunden wurden die beiden mit einer Site-to-Site IPSec-VPN-Verbindung. Ebenso wie im realen Aufbau wurden zwei Server (Betriebssystem: Debian GNU/Linux 8 Jessie, 64 Bit, 256MB RAM) hinter einer jeweiligen Firewall aufgesetzt, welche zum einen den Lohnfertiger-Server und zum anderen den LOGICDATA-Server darstellen.

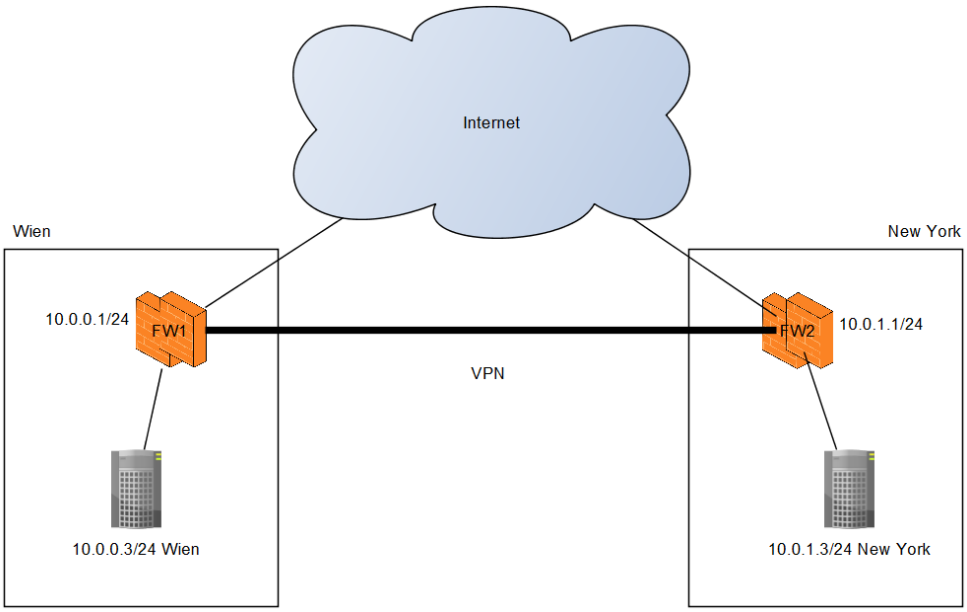


Abbildung 35: Netzwerkdiagramm der Testumgebung, Quelle: Eigene Darstellung.

Für die weitere Beschreibung der praktischen Arbeit sind auch die jeweiligen IP-Adressen der Firewalls und Server in Abbildung 35 ersichtlich.

6.2 Auswahlkriterien

Für die Auswahl einer Monitoring-Software wurden firmenintern Kriterien definiert, die die Suche nach einer geeigneten Lösung eingeschränkt haben. An dieser Stelle ist zu erwähnen, dass sich im Zuge des Evaluierungsprozesses eine Aufteilung der Software-Lösung in zwei Teilbereiche als sinnvoll erwiesen hat. Konkret ist hier gemeint, dass sich ein Monitoring-Tool speziell auf die Bandbreitenmessung konzentriert, und ein zweites hauptsächlich auf Parameter am Server (z.B. CPU-Auslastung, Speicherplatz, etc.). Dies wurde entschieden, da sich keine allumfassende Lösung zu den Anforderungen finden ließ.

6.2.1 Kriterien Bandbreiten-Monitoring-Tool

6.2.1.1 Zuverlässigkeit der Daten

Bevor nun die getesteten Software-Lösungen näher erläutert werden, muss noch der Vorgang der Plausibilitätsprüfung der angezeigten Daten präziser beschrieben werden. Dieser Prozess wurde für jedes Tool gleichermaßen durchgeführt. Das Ergebnis ist in der Übersicht jeder Software unter dem Kriterium „Zuverlässigkeit der Daten“ ersichtlich.

Die Plausibilität der Bandbreitendaten wurde durch einen „iPerf“-Test überprüft (künstlich erzeugte Auslastung). Mit iPerf kann ein definierter Datenstrom generiert werden, welcher auch auf der Firewall ersichtlich ist (siehe Werte „Bandwidth“ in Abbildung 36). Die Daten auf der Firewall (vergleiche Abbildung 37) und die Daten des Monitoring-Tools müssen dabei übereinstimmen. Für den Test müssen Parameter am Sender (Server) und Empfänger (Client) folgendermaßen festgelegt werden:

Empfänger: *iperf3 -s*

Sender (IP-Adresse des Servers + optional die Zeit in Sekunden): *iperf3 –c 10.0.1.3 –t 300*

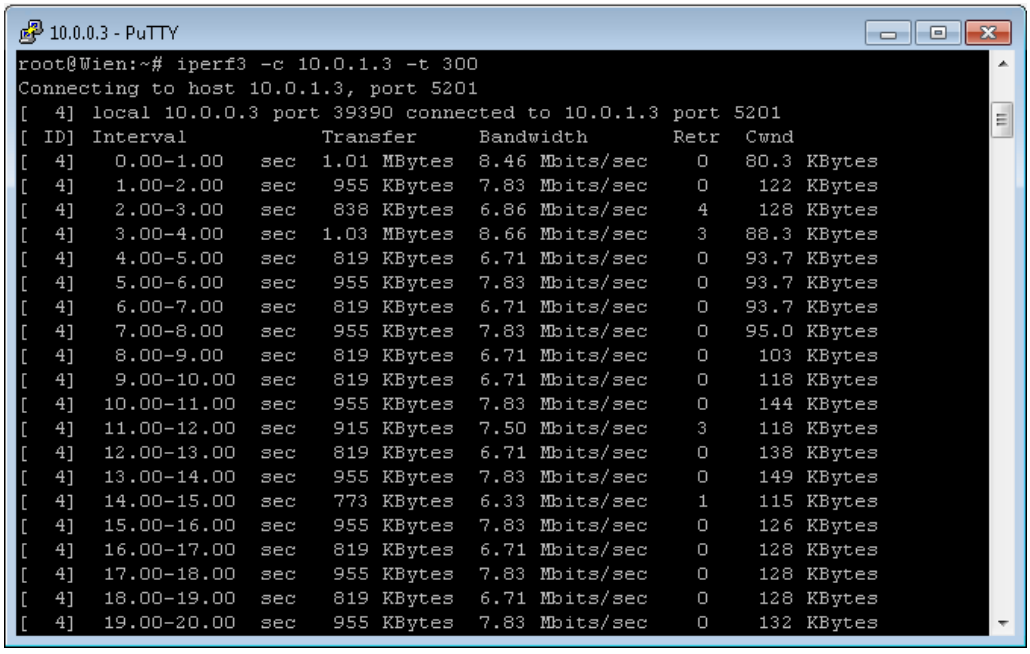


Abbildung 36: Durchführung des iPerf-Tests, Quelle: Eigene Darstellung.

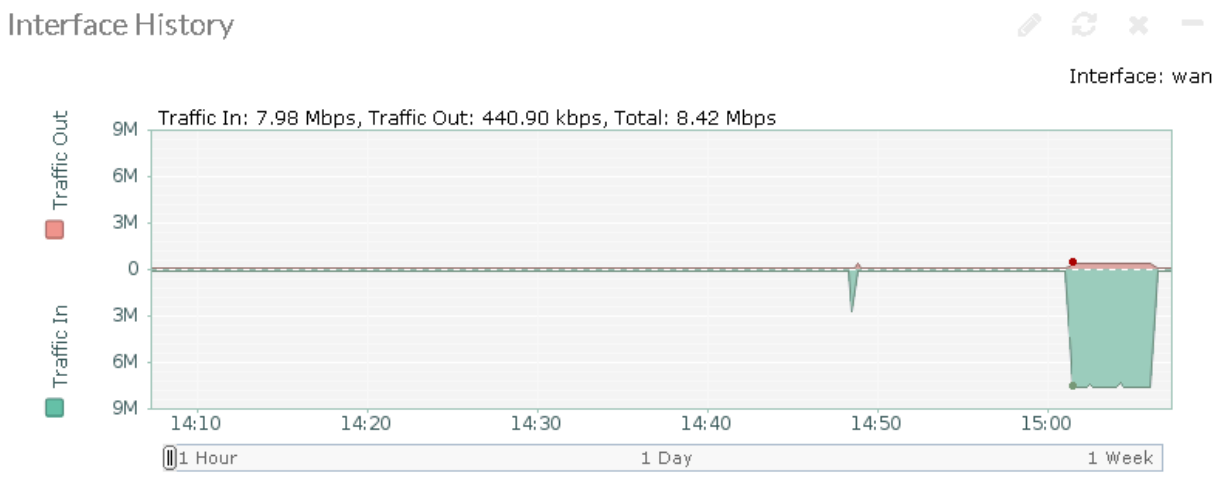


Abbildung 37: Anzeige der Traffic-Daten auf der Firewall, Quelle: Eigene Darstellung.

6.2.1.2 Grafische Darstellung

Um beim alltäglichen Gebrauch der Software möglichst viele Informationen in kurzer Zeit erhalten zu können, ist eine gute grafische Darstellung der Daten Voraussetzung. Ein schneller Überblick über die Verbindungen muss für jeden zuständigen Mitarbeiter gegeben sein. Dieses Kriterium ist natürlich stark vom Anwender abhängig und kann nicht eindeutig festgestellt werden. Es wird aber versucht, eine möglichst treffende Einschätzung auszuarbeiten.

6.2.1.3 Kosten

Dieser Faktor hält fest, welche Kosten auf das Unternehmen bei der Nutzung der Software zukommen. Vorgezogen werden natürlich Lösungen, für die keine Kosten anfallen (Voraussetzung: die anderen Kriterien werden erfüllt).

6.2.1.4 Support von SFlow v5 bzw. NetFlow v9

Wie bereits in Kapitel 5.2.2 „SFlow“ und 5.2.1 „NetFlow“ beschrieben, ist die Unterstützung einer der beiden Technologien unbedingt notwendig, da eine Messung der Bandbreite mit SNMP nicht die Details zur Verfügung stellt, die für eine präzise Auswertung der Bandbreitendaten notwendig sind. Die Versionsvorgaben (Version 5 und Version 9) ergeben sich aus der Vorgabe der verwendeten Firewall. Sowohl die Test- als auch die Produktiv-Firewalls unterstützen nur diese beiden Versionen der Technologien.

6.2.1.5 Live-Monitoring-Fähigkeit

Die gemessenen Daten der Bandbreiten-Monitoring-Software sollten mit möglichst geringer Verzögerung angezeigt werden, um die aktuelle Situation jederzeit bestmöglich einschätzen zu können. Die evaluierten Lösungen wurden bis zu einer Verzögerung von 30 Sekunden als „Live-Monitoring-fähig“ eingestuft. Da eine Installation auf den tatsächlich verwendeten Lohnfertiger-Servern erst nach Abschluss dieser Arbeit erfolgt, wird nun im folgenden Absatz erläutert, wie eine präzise Messung der Anzeigeverzögerung erfolgen könnte.

Mithilfe von Zeitstempeln auf beiden Servern könnte die Differenz (=Anzeigeverzögerung) auf Millisekunden genau festgestellt werden. Davor muss eine Zeit-Synchronisation der beiden Geräte (z.B. mithilfe von NTP[116]) stattfinden, um ein plausibles Ergebnis erzielen zu können. Als Startzeitpunkt der Messung dient ein Logging-Eintrag, welcher beim Auftreten eines bestimmten Ereignisses am Lohnfertiger-Server erstellt werden müsste. Im optimalen Fall stellt die Monitoring-Software ebenfalls Logging-Einträge zur Verfügung, welche dann ausgewertet werden könnten. Sollte das nicht der Fall sein, sind die Logging-Einträge der Firewall heranzuziehen, wodurch sich das Ergebnis minimal verfälschen würde.

6.2.1.6 Filterungsmöglichkeiten

Neben einem schnellen ersten Überblick müssen für eine tiefergehende Analyse (z.B. bei auftretenden Komplikationen) entsprechende Filtermöglichkeiten von der Software zur Verfügung gestellt werden. Es muss nachvollziehbar sein, woher die Daten stammen (Quell- und Zieladressen) und von wem die Bandbreite verursacht wird.

6.2.2 Kriterien Server-Monitoring-Tool

6.2.2.1 „Agentless"-Monitoring-Fähigkeit

Sollte eine Installation der Software auf dem zu überwachenden Server nicht möglich sein (z.B. durch örtliche Gegebenheiten, geringe Leistungsfähigkeit des Servers, etc.), kann eine sogenannte „agentless"-Überwachung Erleichterung bringen. Dafür muss nur am „Überwachungs-Server" die Software installiert werden, von dem aus dann die gewählten Server beobachtet werden. Da die Lohnfertiger-Server weltweit verteilt sind, wäre es vorteilhaft, einen Teil der zu überwachenden Parameter „agentless" überwachen zu können.

6.2.2.2 Kosten

Bereits beschrieben in Kapitel 6.2.1.3 „Kosten".

6.2.2.3 Überprüfbare Parameter

Ist die Software in der Lage, wichtige Parameter, die Aufschluss über den Status des Servers geben (z.B. Erreichbarkeit, Speicherplatz, CPU-Auslastung, etc.), zu überwachen?

6.2.2.4 Benachrichtigungen

Im kritischen Fall sollen Benachrichtigungen an verantwortliche Personen gesendet werden. Dies soll vorzugsweise über Email erfolgen. Eine zusätzliche SMS-Funktion wäre von Vorteil, ist aber nicht zwingend notwendig.

6.2.2.5 Grafische Darstellung

Bereits beschrieben in Kapitel 6.2.1.2 „Grafische Darstellung".

[116] NTP…Network Time Protocol.

6.3 Überblick der getesteten Lösungen

Nach Definition der oben genannten Kriterien wurde der Suchprozess für geeignete Monitoring-Lösungen gestartet. In den folgenden Kapiteln werden nun die untersuchten Lösungen für die jeweiligen Teilbereiche kurz vorgestellt.

6.3.1 Überblick Bandbreiten-Monitoring-Tools

Für die Überwachung der Bandbreite wurden sechs verschiedene Lösungen getestet. Konkret wurde auf der Firewall das Interface untersucht, auf welchem der VPN-Traffic zwischen den Testmaschinen „New York“ und „Wien“ sichtbar ist.

Bevor aber ein (SFlow- oder NetFlow)-Kollektor Daten empfangen kann, müssen diese erst von der Firewall exportiert werden. Die verwendete Firewall unterstützt die Version 5 von SFlow und 9 von NetFlow. Dies wurde folgendermaßen in der Kommandozeile (CLI) auf der Firewall konfiguriert:

Konfiguration SFlow	Konfiguration NetFlow

`config system sflow`
`set collector-ip 10.0.0.3`
`set collector-port 6343`
`set source-ip 10.0.1.1`
`end`
`config system interface`
`edit wan`
`set sflow-sampler enable`
`set sample-rate 1024`
`set sample-direction both`
`set polling-interval 30`
`next`
`end`
	`config system netflow`
	`set collector-ip 192.168.15.127`
	`set collector-port 2055`
	`set source-ip 10.0.0.1`
	`set active-flow-timeout 1`
	`set inactive-flow-timeout 15`
	`end`
	`config system interface`
	`edit wan1`
	`set netflow-sampler both`
	`end`

Testen der Einstellungen: `get system sflow`	**Testen der Einstellungen:** `get system netflow`

Tabelle 9: SFlow- & NetFlow-Konfiguration auf der Firewall.

Für einen ersten Test, ob die geänderten Einstellungen übernommen wurden, wurde eine Test-Software der Paessler AG (jeweils eine für SFlow und eine für NetFlow) eingesetzt. Diese zeigt an, ob der eingestellte Kollektor-Rechner auch tatsächlich SFlow- bzw. NetFlow-Pakete empfängt (siehe Abbildung

38 und Abbildung 39). Für das weitere Vorgehen im Evaluierungsprozess war diese Information sehr hilfreich.

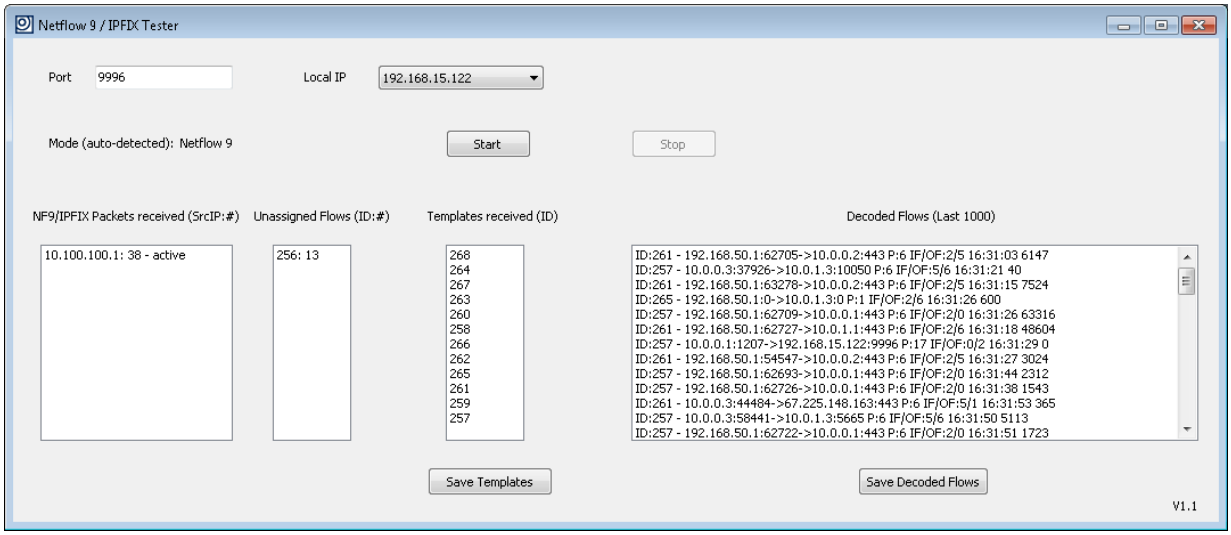


Abbildung 38: Netflow 9 Tester, Quelle: Eigene Darstellung.

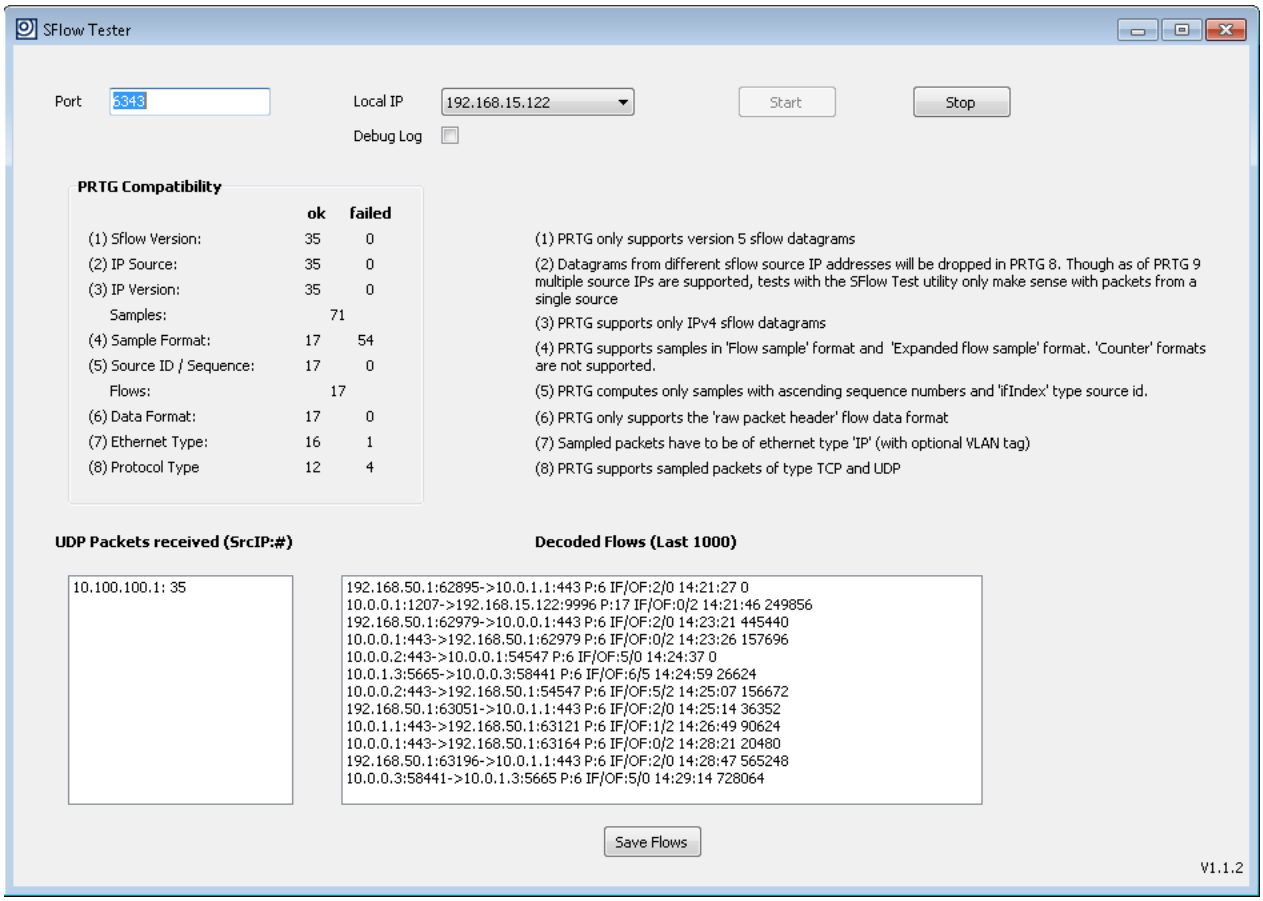


Abbildung 39: SFlow 5 Tester, Quelle: Eigene Darstellung.

6.3.1.1 NetFlow Realtime Analyzer

Overview „NetFlow Realtime Analyzer Version 10.6.1“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein	Support von Sflow v5	Wird nicht unterstützt
Grafische Darstellung	Gut	Support von NetFlow v9	Wird nicht unterstützt
Kosten	Kostenlos	Live-Monitoring-Fähigkeit	Ja
Filtermöglichkeiten	Unzureichend	Getestet auf Betriebs-system	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 48MB, CPU-Auslastung: 0,14%		

Tabelle 10: Overview "NetFlow Realtime Analyzer".

Der Analyzer der Fa. Solarwinds bietet eine einfache Installation und Konfiguration und kann den ein- und ausgehenden Traffic für die einzelnen Interfaces auf der Firewall anzeigen. Die Installationsdatei steht unter www.solarwinds.com zum Download bereit. Die Plausibilität der Daten durch Vergleich mit der Firewall wurde ebenfalls durchgeführt. Allerdings wird eine grafische Auswertung der Netflow-Pakete nur bis zur Netflow-Version 5 bereitgestellt. Da die Firewall nur Netflow v9 unterstützt, konnte dieses Tool nicht vollständig getestet werden und ist auch für einen weiteren Einsatz ungeeignet.

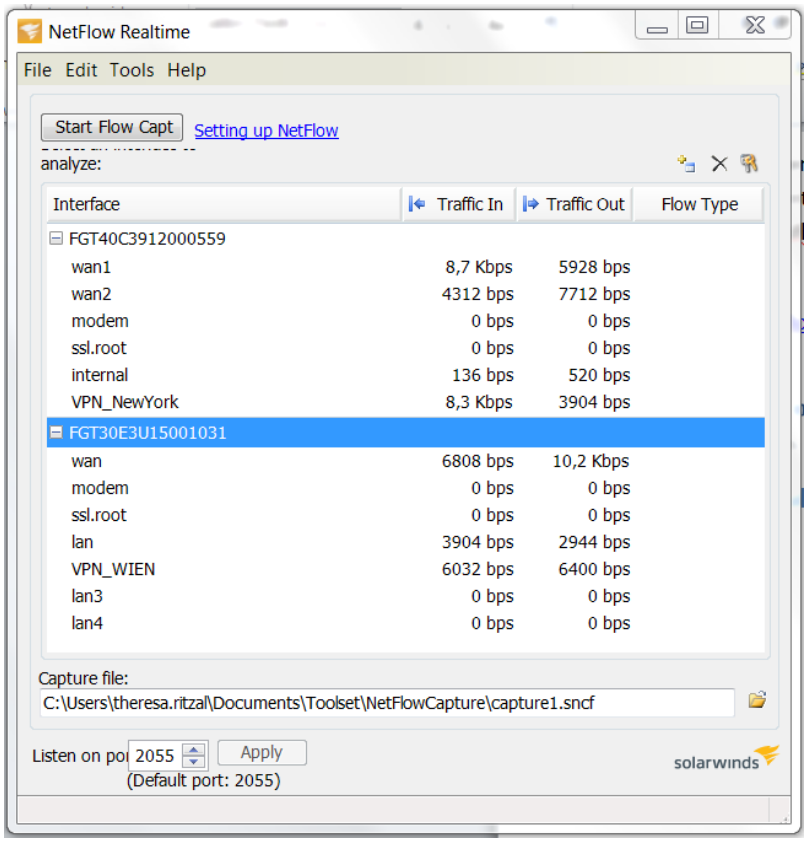


Abbildung 40: Benutzeroberfläche "NetFlow Realtime Analyzer", Quelle: Eigene Darstellung.

6.3.1.2 FlowAlyzer

Overview „FlowAlyzer Version 2.0“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein (SNMP)	Support von Sflow v5	Wird unterstützt
Grafische Darstellung	Unzureichend	Support von NetFlow v9	Wird unterstützt
Kosten	Kostenlos	Live-Monitoring-Fähigkeit	Ja
Filtermöglichkeiten	Keine	Getestet auf Betriebs-system	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 282MB, CPU-Auslastung: 0,33%		

Tabelle 11: Overview "FlowAlyzer".

Das Tool, dessen Installationsdatei unter www.plixer.com zu finden ist, ist gut geeignet um festzustellen, ob das Kollektor-Gerät überhaupt Netflow bzw- Sflow-Pakete empfängt. Einen grafischen Verlauf dieser Flow-Daten zu erstellen ist allerdings nicht möglich (siehe Abbildung 41). Des Weiteren kann nicht nach der Quell-IP-Adresse gefiltert werden. Nur der zuletzt passierte Router wird angezeigt („Exporter IP“). Eine Darstellung des Traffics ist nur mithilfe von SNMP möglich (unter dem Reiter „Trender“, siehe Abbildung 42).

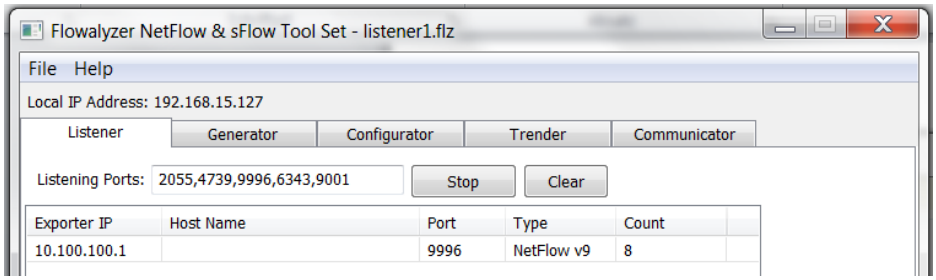


Abbildung 41: Benutzeroberfläche "FlowAlyzer", Quelle: Eigene Darstellung.

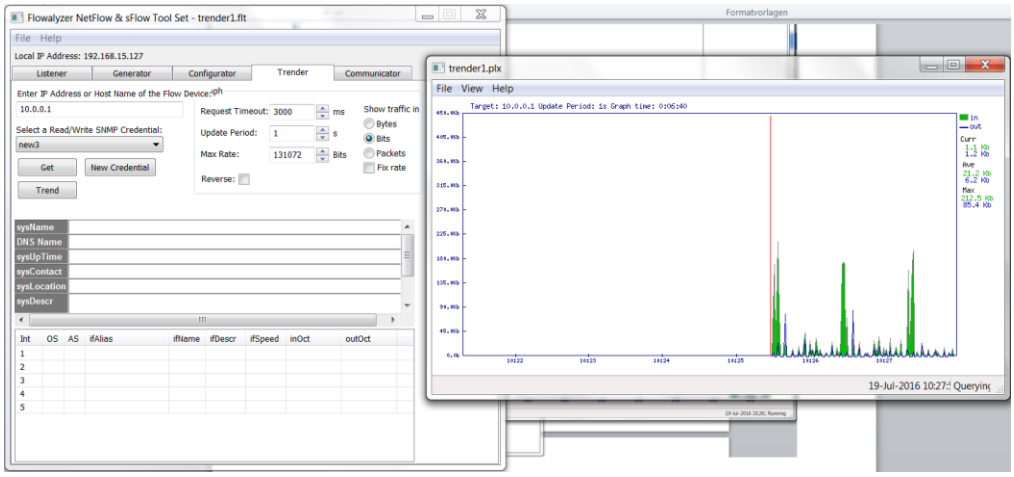


Abbildung 42: Darstellung des Traffics mittels SNMP, Quelle: Eigene Darstellung.

6.3.1.3 Fireplotter

Overview „Fireplotter Version 2.24“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein	Support von Sflow v5	Wird nicht unterstützt
Grafische Darstellung	Ausreichend aber ausbaufähig	Support von NetFlow v9	Wird nicht unterstützt
Kosten	Kostenlos für Standard-Services (z.B. HTTP, FTP, SMTP, etc.), im erweiterten Modus kostenpflichtig	Live-Monitoring-Fähigkeit	Ja
Filtermöglichkeiten	Ausreichend aber ausbaufähig	Getestet auf Betriebs-system	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 2,8MB, CPU-Auslastung: 0,62%		

Tabelle 12: Overview "Fireplotter".

Die Software „Fireplotter“ ist an sich kein SFlow- oder NetFlow-Kollektor, sondern stellt Daten zu den gerade ablaufenden Sessions dar. Getestet wurde die Version 2.24 (Build 160916), welche unter www.fireplotter.com zum Download bereitsteht. Eine ausreichende Übersicht über die Bandbreite ist für eine schnelle Überprüfung dennoch gegeben (siehe Abbildung 43). Für ein tiefer gehendes Monitoring ist die Darstellung der Daten nicht geeignet, da nur ein Zeitraum von 20 Minuten angezeigt und eine Gesamtdarstellung der einkommenden und ausgehenden Daten nicht zur Verfügung gestellt wird. Ein weiterer Faktor stellt die Filterung dar, die zwar grundsätzlich die Möglichkeit zu einer besseren Übersicht gibt (z.B. Filterung nach Ziel- und Quell-Adresse), allerdings das Diagramm (je nach Filterung) nicht verändert. Es werden immer alle Daten im Diagramm angezeigt. Hervorzuheben ist die einfache Installation und Inbetriebnahme der Software, die kurzen Abstände zwischen den Datenupdates (ca. alles 5 Sekunden) und die Darstellung der Daten verschiedener Typen in verschiedenen Farben (siehe Abbildung 43).

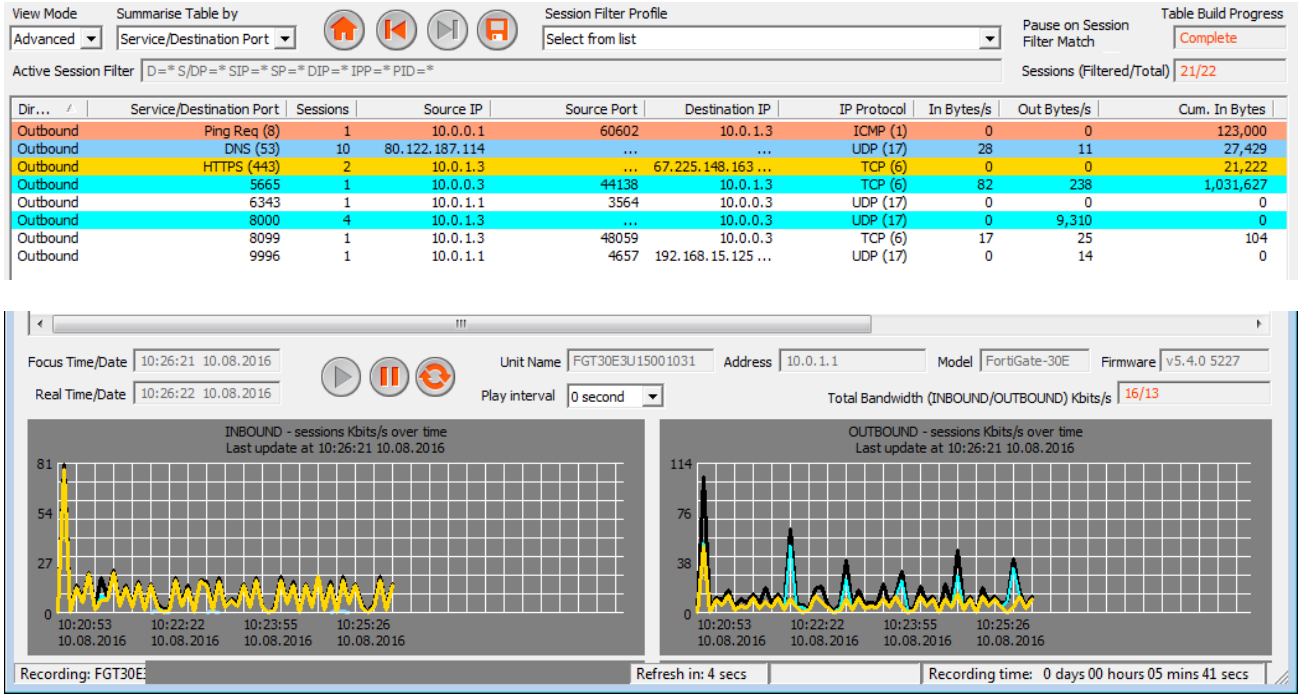


Abbildung 43: Traffic-Tabelle und Darstellung der Bandbreite im "Fireplotter", Quelle: Eigene Darstellung.

6.3.1.4 PRTG

Overview „PRTG Version 16.3.26.6384“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein	Support von Sflow v5	Wird unterstützt
Grafische Darstellung	Sehr gut	Support von NetFlow v9	Wird unterstützt
Kosten	Kostenlos bis 100 Sensoren	Live-Monitoring-Fähigkeit	Nein (ca. 15 Minuten Verzögerung)
Filtermöglichkeiten	Nicht ausreichend	Getestet auf Betriebs-system	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 33MB, CPU-Auslastung: 4,12%		

Tabelle 13: Overview "PRTG".

Die von der Firma „Paessler“ (www.de.paessler.com) entwickelte Monitoring-Software „PRTG“ (getestet in der Freeware-Version, auf 100 Sensoren limitiert) bietet nicht nur Funktionalitäten zur Bandbreitenüberwachung, sondern auch sehr viele Möglichkeiten zum Server-Monitoring. Darauf wird in Kapitel 6.3.2 „Überblick Server-Monitoring-Tools“ weiter eingegangen. PRTG ist das einzig getestete Tool, das Netflow v9 unterstützt und die empfangenen Daten auf den ersten Blick sehr anschaulich darstellt (siehe Abbildung 44). Der Nachteil ist hierbei aber, dass die Filterung nach der Quelladresse nicht benutzerfreundlich und somit eine Nachvollziehbarkeit der Daten nur schwer möglich ist. Außerdem wurde eine willkürliche Aktualisierung der Daten (kein fixes Intervall) im Zuge der Testphase festgestellt.

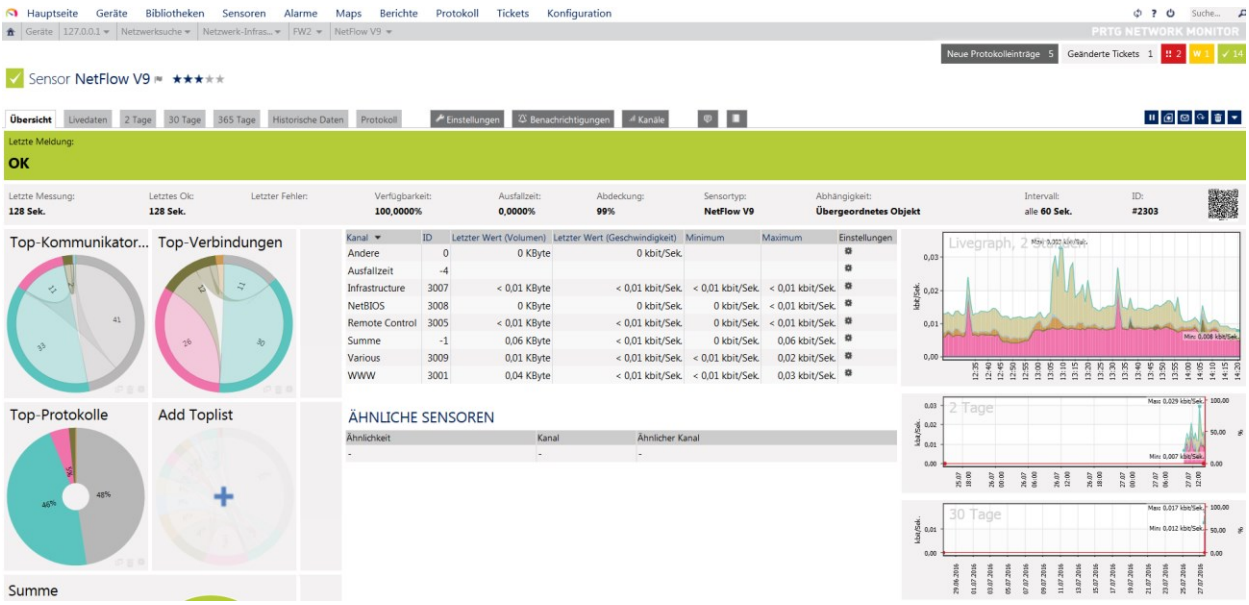


Abbildung 44: Benutzeroberfläche "PRTG".

6.3.1.5 Solarwinds RealTime Bandwidth Monitor

Overview „Solarwinds RealTime Bandwidth Monitor Version 1.0.0.114“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein	Support von Sflow v5	Wird nicht unterstützt
Grafische Darstellung	Sehr gut	Support von NetFlow v9	Wird nicht unterstützt
Kosten	Kostenlos	Live-Monitoring-Fähigkeit	Ja
Filtermöglichkeiten	Keine	Getestet auf Betriebs-system	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 78MB, CPU-Auslastung: 0,11%		

Tabelle 14: Overview "Solarwinds RealTime Bandwidth Monitor".

Die Software „RealTime Bandwidth Monitor“ wurde ebenso von Solarwinds (Download des aktuellen Programms unter: www.solarwinds.com) entwickelt und bietet eine Alternative zum „NetFlow Realtime Analyzer“. Das Tool bietet nicht die Funktionalität eines NetFlow-Kollektors, sondern kann den Traffic über SNMP auf einem ausgewählten Interface abfragen und mithilfe einer guten grafischen Darstellung einen schnellen Überblick verschaffen. Dies ist besonders leicht durch die einfache Installation und Konfiguration möglich.

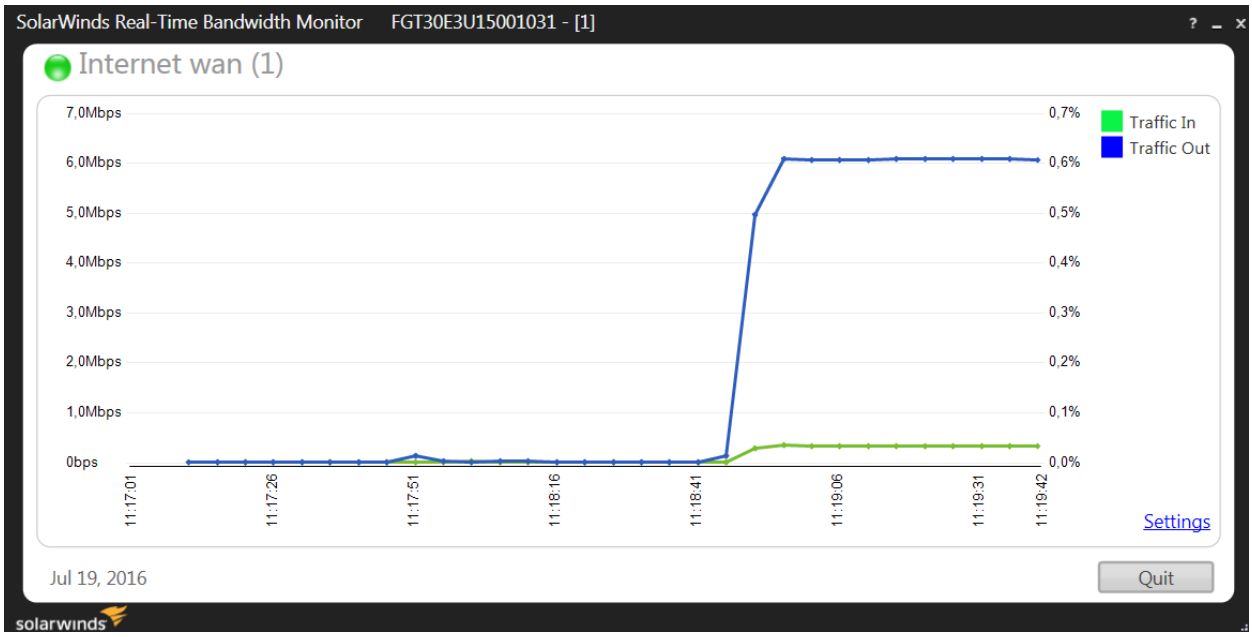


Abbildung 45: Darstellung des Traffics durch "RealTime Bandwidth Monitor", Quelle: Eigene Darstellung.

6.3.1.6 SFlow Trend

Overview „SFlow Trend Version 6.3.01“			
Zuverlässigkeit der Daten	Stimmen mit den Werten auf der Firewall überein	Support von Sflow v5	Wird unterstützt
Grafische Darstellung	Sehr gut	Support von NetFlow v9	Wird nicht unterstützt
Kosten	Kostenlos	Live-Monitoring-Fähigkeit	Ja
Filtermöglichkeiten	Sehr gut	Getestet auf Betriebs-system	Debian GNU/Linux 8 Jessie, 64 Bit, 1GB RAM
Ressourcenbedarf	RAM: 61MB, CPU-Auslastung: 0,33%		

Tabelle 15: Overview "SFlow Trend".

„SFlow Trend“ ist ein klassischer (und kostenloser) SFlow-Kollektor, welcher am Betriebssystem Debian Jessie getestet wurde. Die aktuellen Installationsdateien sind unter www.inmon.com aufgelistet und können, je nach verwendetem Betriebssystem, heruntergeladen werden. Die Installation und Konfiguration ist zwar etwas aufwendiger als bei Windows-Lösungen, kann aber durch andere Vorzüge überzeugen. Die grafische Darstellung der Bandbreite und die Filtermöglichkeiten sind sehr benutzerfreundlich ausgerichtet (siehe Abbildung 46).

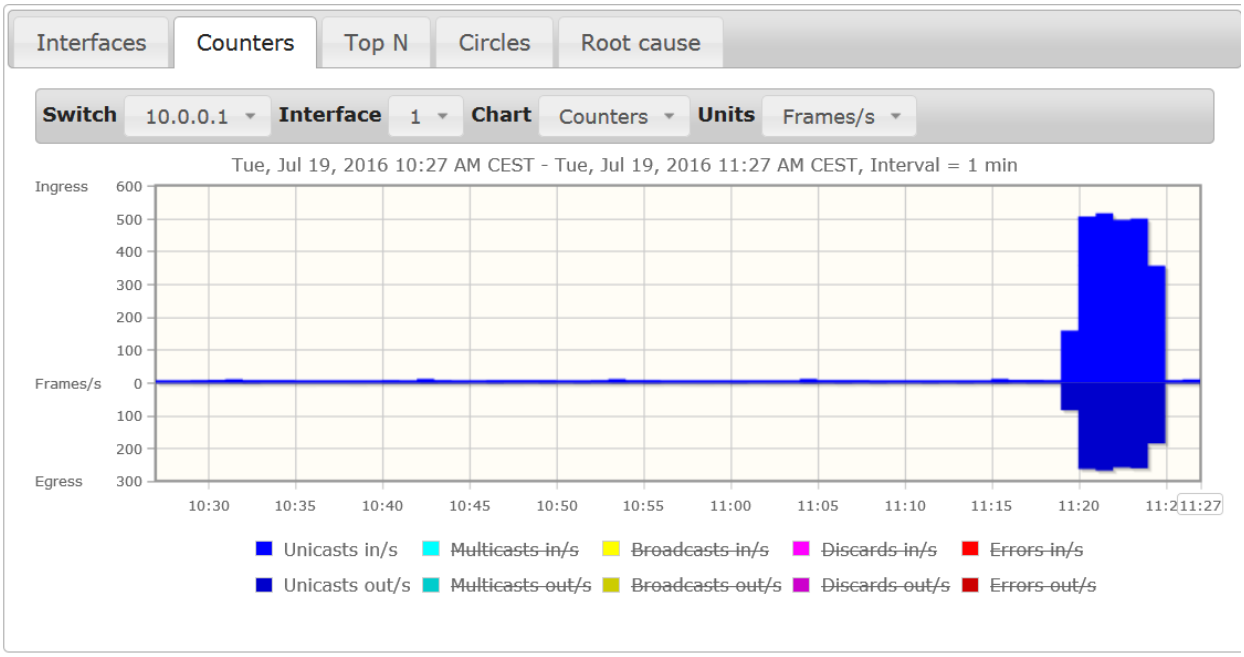


Abbildung 46: Darstellung des Traffics durch „SFlow Trend“, Quelle: Eigene Darstellung.

6.3.2 Überblick Server-Monitoring-Tools

Die im folgenden Kapitel beschriebenen Software-Lösungen sind spezialisiert auf das Überwachen von Servern bzw. auch anderen Netzwerkkomponenten. Vier verschiedene Tools wurden im Zuge dieser Masterarbeit evaluiert.

6.3.2.1 Icinga

Overview „Icinga Version 2.3.4“	
Agentless-Monitoring-Fähigkeit	Teilweise (für HTTP, SSH, Hostalive (PING))
Kosten	Kostenlos
Überprüfbare Parameter	HTTP-Service, SSH-Service, Erreichbarkeit des Agents (PING), verfügbare Updates, Speicherplatz auf Laufwerk, Auslastung des Systems, freier SWAP-Platz, Anzahl der Prozesse, eingeloggte Benutzer
Benachrichtigungen	E-Mail, Twitter, etc. (mithilfe zusätzlicher Add-Ons)
Grafische Darstellung	Sehr gut
Getestet auf Betriebssystem	Debian GNU/Linux 8 Jessie, 64 Bit, 256MB RAM
Ressourcenbedarf	RAM: 2,4MB, CPU-Auslastung: 0,43%

Tabelle 16: Overview "Icinga".

Die Software „Icinga“ basiert auf der bereits bewährten, aber käuflich zu erwerbenden Software „Nagios“. Icinga wurde von einer Reihe erfahrener Entwickler weiterentwickelt. Die Installationspakete sind unter www.icinga.org für diverse Betriebssystem-Typen verfügbar. Die OpenSource-Lösung bietet eine große

Anzahl an Möglichkeiten zur Überwachung der Lohnfertiger-Server (HTTP, SSH und PING können sogar „agentless“ überwacht werden), welche die Daten im Hintergrund in einer MySQL-Datenbank verwaltet. Durch die Installation und Konfiguration mittels eines Kommandozeilen-Interfaces (nicht von der Web-Oberfläche aus) ist zwar ein höherer Einarbeitungsaufwand notwendig, muss aber für den Grad an Flexibilität in Kauf genommen werden. Außerdem wird dem Benutzer eine sehr umfangreiche Online-Dokumentation zur Verfügung gestellt, welche bei der Installation der Software unterstützt.

Icinga bietet die Möglichkeit eines hierarchischen Aufbaus. Dabei werden sogenannte zentrale „Master“ definiert, welche definierte „Satellites“ überwachen. Auf dem „Master“ ist sowohl die Web-Applikation, als auch die zentrale Datenbank installiert. Abbildung 47 ermöglicht einen ersten Einblick in das „Dashboard“ der Monitoring-Software.

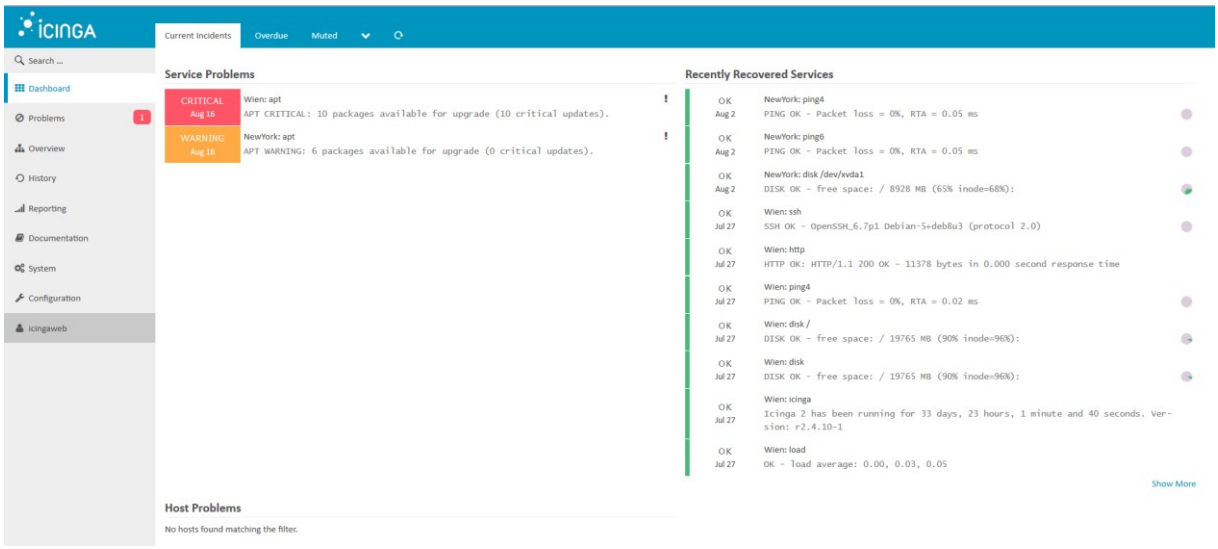


Abbildung 47: Web-Oberfläche von "Icinga", Quelle: Eigene Darstellung.

6.3.2.2 PRTG

Overview „PRTG Version 16.3.26.6384“	
Agentless-Monitoring-Fähigkeit	Ja
Kosten	Kostenlos bis 100 Sensoren (Abfragen auf Geräten)
Überprüfbare Parameter	Verfügbarkeit, Bandbreite (SNMP und NetFlow/SFlow), Geschwindigkeit, Prozessornutzung, Datenträgernutzung, Speichernutzung
Benachrichtigungen	E-Mail, SMS, Eintrag in Windows-Logs
Grafische Darstellung	Sehr gut
Getestet auf Betriebssystem	Windows 7, 64 Bit, SP 1, 8GB RAM
Ressourcenbedarf	RAM: 33MB, CPU-Auslastung: 4,12%

Tabelle 17: Overview "PRTG".

Wie bereits in Kapitel 6.3.1.4 beschrieben, kann die Software „PRTG“ nicht nur für die Bandbreiten-überwachung genutzt werden, sondern auch für Server-Monitoring. Die Überwachung findet hier mit sogenannten „Sensoren“ statt, die einen Monitoring-Dienst beschreiben (z.B. PING, HTTP, etc.). Diese Sensoren werden bereits fertig zur Verfügung gestellt (etwa 200), die vom Benutzer nur mehr ausgewählt und konfiguriert werden müssen. Dies erleichtert zwar den Einstieg, ist aber auf bei längerer Nutzung unpraktisch, da man durch die vorgefertigten Sensoren unflexibel in der Anwendung ist (keine Änderung von Basis-Einstellungen).

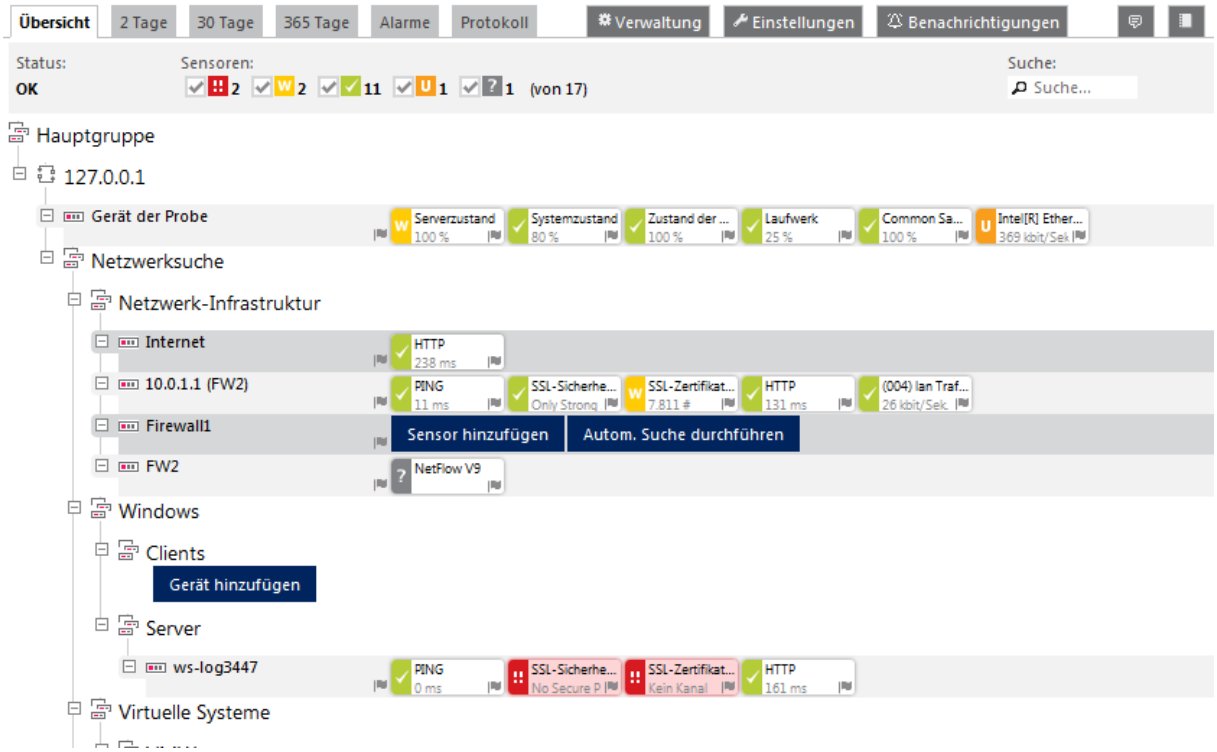


Abbildung 48: Benutzeroberfläche von "PRTG", Quelle: Eigene Darstellung.

6.3.2.3 Anturis

Overview „Anturis Version 2.5“	
Agentless-Monitoring-Fähigkeit	Ja
Kosten	Ab $7,50/Monat (Überwachung von 10 Geräten)
Überprüfbare Parameter	Bandbreite (SNMP), Prozessornutzung, Festplattennutzung, Möglichkeit der Erweiterung um eigene Parameter
Benachrichtigungen	E-Mail, SMS und Voice-Call-Notifications
Grafische Darstellung	Ausreichend aber ausbaufähig
Getestet auf Betriebssystem	„Software as a Service“
Ressourcenbedarf	Keine, da keine lokale Installation stattfindet

Tabelle 18: Overview "Anturis".

Das Tool „Anturis“ wird als SaaS[117] angeboten, das neben diversen Servern auch Websiten, Datenbanken, Netzwerkverbindungen und sogar das ActiveDirectory überwachen kann. Die Software muss nicht mehr lokal installiert werden, eine einfache Registrierung genügt, um einen Zugang zum Web-Portal zu bekommen (siehe Abbildung 49). Anturis überzeugt zwar durch die einfache Handhabung, die Grenzen der Monitoring-Möglichkeiten sind allerdings schnell erreicht. Z.B. stehen für die Überwachung eines Servers standardmäßig nur die Überwachung der CPU und der Laufwerke zur Verfügung, deren grafische Darstellung sich auf das Mindeste konzentriert.

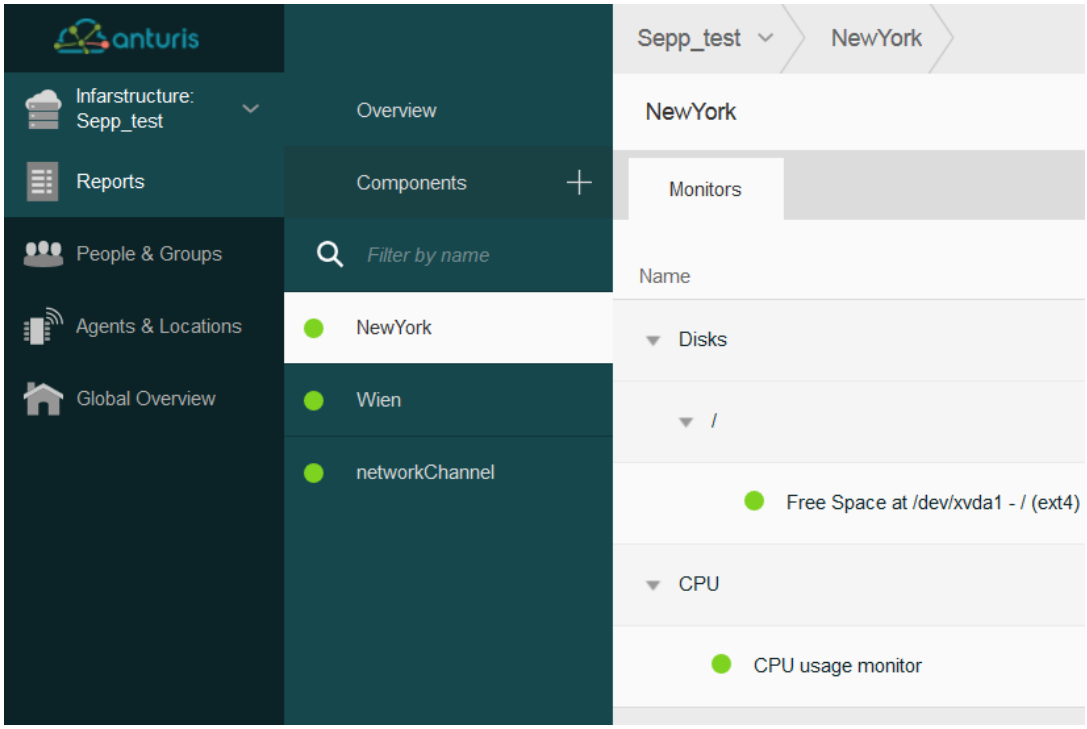


Abbildung 49: Benutzeroberfläche "Anturis", Quelle: Eigene Darstellung.

[117] SaaS…Software as a Service.

7 EINFÜHRUNG EINER AUSGEWÄHLTEN MONITORING-SOFTWARE

Eine Reihe von unterschiedlichen Monitoring-Tools (sowohl für den Server- als auch für den Bandbreiten-Bereich) wurde installiert und getestet. Das folgende Kapitel widmet sich im ersten Teil dem Vergleich der bereits oben erwähnten Lösungen. Die Ergebnisse daraus werden zu einer Auswahl jeweils einer Bandbreiten- und Server-Monitoring-Software führen. Die Installation und Konfiguration dieser in der Testumgebung wird im zweiten Teil dieses Kapitels behandelt.

7.1 Ergebnisse aus Vergleich

Der Vergleich wird nun wieder in Bandbreiten- und Server-Monitoring-Tools unterteilt, um einen guten Überblick über die Einzelbereiche zu erhalten.

7.1.1 Vergleich Bandbreiten-Monitoring-Tools

Ein weiteres ausschlaggebendes Kriterium konnte im Evaluierungsprozess festgemacht werden – nämlich die Verwendung von NetFlow oder SFlow (statt SNMP) für die Bandbreitenmessung. Dies kann damit begründet werden, dass mithilfe von SNMP zwar die korrekten Bandbreitenwerte anzeigt werden (und somit eine grobe Übersicht gegeben ist), allerdings keine weiteren Details zur Verfügung stehen. Diese wiederum stellen NetFlow und SFlow bereit. Hier kann genau festgestellt werden, von welchem Endsystem die Bandbreite benötigt wird. Diese Information kann oftmals bei Engpässen behilflich sein.[118]

Im folgenden Abschnitt wird nun der direkte Vergleich der Software-Lösungen für Bandbreiten-Monitoring vorgenommen:

Name	Vorteile	Nachteile	Resümee
NetFlow Realtime Analyzer	- Einfache Installation - Kostenlos - Live-Monitoring möglich	- Weder NetFlow v9 noch SFlow v5 werden unterstützt	Kommt nicht für einen Einsatz in Frage, da keine der benötigten Versionen unterstützt wird
Flowalyzer	- Korrekte Werteanzeige - Kostenlos - Unterstützung von NetFlow v9 und SFlow v5 - Einfache Installation	- Unzureichende grafische Darstellung (nur für SNMP) - Keine Filtermöglichkeiten	Für Tests sehr praktikabel, für einen Monitoring-Einsatz eher ungeeignet, da NetFlow und SFlow nicht ausreichend unterstützt werden
Fireplotter	- Korrekte Werteanzeige - Kostenlos für Standard-	- Weder NetFlow v9 noch SFlow v5 werden unter-	Kommt nicht für einen Einsatz in

[118] Vgl. Patterson (2010), Online-Quelle [5.September.2016].

	Services - Einfache Installation	stützt - Ausbaufähige grafische Darstellung - Suboptimale Filtermöglichkeiten	Frage, da keine der benötigten Versionen unterstützt wird
PRTG	- Korrekte Werteanzeige - Unterstützung von NetFlow v9 und SFlow v5	- Kein Live-Monitoring möglich - Filtermöglichkeiten nicht ausreichend	Sehr mächtiges Tool; durch die fehlenden Filtermöglichkeiten und die Verzögerung bei der Anzeige der Daten, kommt PRTG für einen Einsatz nicht in Frage
Solarwinds Realtime Bandwidth Monitor	- Sehr gute grafische Darstellung - Kostenlos - Live-Monitoring möglich - Einfache Installation	- Weder NetFlow v9 noch SFlow v5 werden unterstützt (nur SNMP) - Keine Filtermöglichkeiten	Scheidet durch die fehlende Unterstützung von NetFlow und SFlow aus; Monitoring nur über SNMP möglich
SFlow Trend	- Korrekte Werteanzeige - Unterstützung von SFlow v5 - Sehr gute grafische Darstellung - Kostenlos - Gute Filtermöglichkeiten - Live-Monitoring möglich	- Keine Unterstützung von NetFlow v9 - Aufwändigere Installation	Für den Einsatz bei LOGICDATA ausgewählt

Tabelle 19: Direkter Vergleich Bandbreiten-Monitoring-Tools.

Wie aus Tabelle 19 ersichtlich, wurde die Software „SFlow Trend“ für einen Einsatz bei LOGICDATA ausgewählt. Die etwas aufwändigere Installation (im Vergleich zu den anderen getesteten Tools) muss zwar in Kauf genommen werden, das kostenlose Tool überzeugt allerdings durch die sehr gute grafische Darstellung, die guten Filtermöglichkeiten und vor allem durch die Unterstützung von SFlow v5. Die Installation und Konfiguration von „SFlow Trend“ werden in Kapitel 7.2.1 beschrieben.

7.1.2 Vergleich Server-Monitoring-Tools

Zunächst werden wieder die einzelnen Software-Lösungen für Server-Monitoring gegenübergestellt und verglichen:

Name	Vorteile	Nachteile	Resümee
Icinga	- Sehr gute grafische Darstellung - Kostenlos - Hohes Maß an Flexibilität - Agentless-Monitoring teilweise möglich - Zur Verfügung stehende Parameter sind ausreichend - Ausgezeichnete Online-Dokumentation	- Aufwändige Installation - Hoher Einarbeitungsaufwand	Für den Einsatz bei LOGICDATA ausgewählt
PRTG	- Agentless-Monitoring möglich - Sehr gute grafische Darstellung - Hohe Anzahl an fertig konfigurierten „Sensoren“ - Einfache Installation - Leichter Einstieg möglich	- Geringer Grad an Flexibilität (durch vorgegebene Sensoren) - Kostenpflichtig (ab 100 Sensoren)	Durch die dauerhafte Einschränkung der fertigen Sensoren (und der anfallenden Kosten) wurde die Software für einen Einsatz abgelehnt
Anturis	- Agentless-Monitoring möglich - Keine Installation nötig (SaaS) - Einfache Handhabung	- Ausbaufähige grafische Darstellung - Wenig Auswahl an zu überwachenden Parametern - Wenig Flexibilität	Anturis bietet zu wenig Möglichkeiten für ein erweiterbares und flexibles Monitoring

Tabelle 20: Direkter Vergleich Server-Monitoring-Tools.

Beim Vergleich der Server-Monitoring-Tools konnte keine Software so eindeutig für den Einsatz bestimmt werden wie bei es bei „SFlow Trend“ der Fall war. Durch die weitgehende Ähnlichkeit der Tools spielten in der Entscheidungsphase auch die Benutzeroberfläche und die intuitive Bedienung eine große Rolle. Icinga besticht insgesamt durch die vielen Überwachungs-Möglichkeiten von Servern (viele Konfigurationsmöglichkeiten), eine benutzerfreundliche grafische Oberfläche und Stabilität. Durch die

ausgezeichnete Online-Dokumentation (und diversen Forumsberichten) konnte der Einarbeitungsaufwand bei der Installation und Konfiguration vermindert werden.

7.2 Installation & Konfiguration der ausgewählten Tools

In den folgenden zwei Kapiteln werden nun die Installation und die Konfiguration der ausgewählten Tools am Testsystem beschrieben.

7.2.1 Installation & Konfiguration von „SFlow Trend“

Aus den Gründen, die in Tabelle 19 genannt wurden, konnte „SFlow Trend“ als das am besten geeignetste Tool für LOGICDATA ausgemacht werden. Beim Aufruf von http://10.0.0.3:8087/sflowtrend/ gelangt man zum Webinterface von „SFlow Trend“. Die Software wurde auf einem Debian-Server (Debian GNU/Linux 8 Jessie) installiert. Mithilfe der gut beschriebenen Online-Dokumentation wurde die Installation gestartet.

7.2.1.1 Installation

Vor der Installation der tatsächlichen Installationsdatei, muss die aktuelle Java-Version installiert werden.

```
apt-get install default-jre-headless
```

Danach wird das Debian-Package von der „inMon“[119]-Homepage heruntergeladen und installiert. Dies erfolgt mithilfe folgender Kommandozeilenbefehle (siehe auch Abbildung 50):

```
wget http://www.inmon.com/products/sFlowTrend/downloads/sFlowTrend-linux-6_3_01.deb
```

```
dpkg -i sFlowTrend-linux-6_3_01.deb
```

```
root@NewYork:/temp# dpkg -i sFlowTrend-linux-6_3_01.deb
(Lese Datenbank ... 71000 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von sFlowTrend-linux-6_3_01.deb ...
Entpacken von sflowtrend (6.3.01) über (6.3.01) ...
sflowtrend (6.3.01) wird eingerichtet ...
insserv: warning: script 'K01nfsen' missing LSB tags and overrides
insserv: warning: script 'nfsen' missing LSB tags and overrides
Starting sflowtrend-server
root@NewYork:/temp#
```

Abbildung 50: Installation von "SFlow Trend" (Auszug aus der Kommandozeile), Quelle: Eigene Darstellung.

Bereits danach kann auf das Web-Interface unter http://10.0.0.3:8087/sflowtrend/ zugegriffen werden. Vor der Konfiguration muss der Benutzer noch angeben, ob er die freie oder kostenpflichtige Version („SFlow Trend-Pro“) von „SFlow Trend“ verwenden möchte (siehe Abbildung 51).

[119] inMon…Die Firma entwickelte SFlow und stellt auch die Software „SFlow Trend“ zur Verfügung.

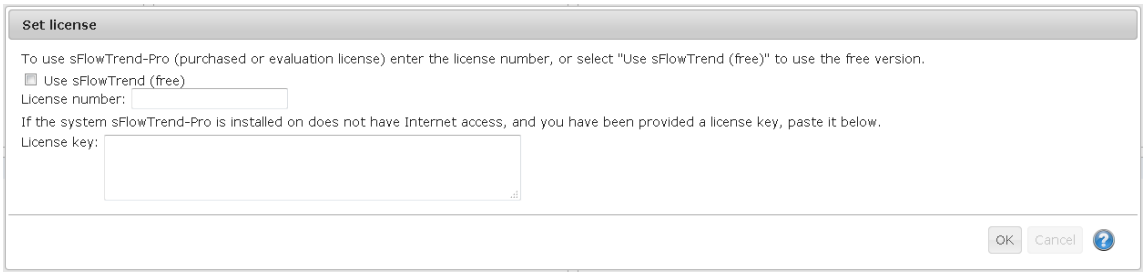


Abbildung 51: Angabe der verwendeten Lizenzform, Quelle: Eigene Darstellung.

Als Startseite wird der Menüpunkt „**Dashboard**“ angezeigt, welcher eine Übersicht über den aktuellen Status des Netzwerks bietet. Es wurde ein Standardbenutzer angelegt.

Vor der Verwendung der Software als SFlow-Kollektor müssen noch die SFlow-Einstellungen, wie in Abbildung 52 ersichtlich, festgelegt werden (Kollektor-Adresse und Port).

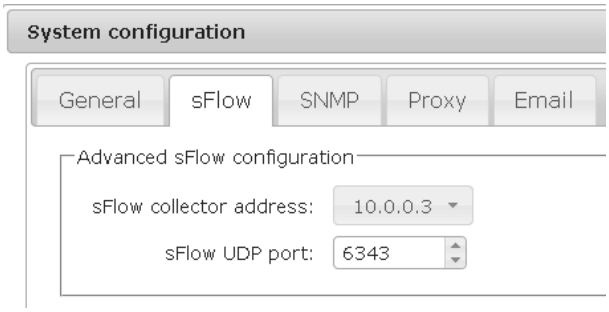


Abbildung 52: Festlegung der SFlow-Einstellungen, Quelle: Eigene Darstellung.

7.2.1.2 Hinzufügen von Agents

Unter den Einstellungen können nun „Agents“ hinzugefügt werden, die überwacht werden sollen (siehe Abbildung 53). Zu Testzwecken wurden die beiden Firewalls hinzugefügt, auf denen bereits die SFlow-Konfiguration erfolgte. Dabei muss bei Erweiterung eines Hosts lediglich die verwendete IP-Adresse und SNMP-Einstellungen (optional) angegeben werden. Wie in Abbildung 54 ersichtlich, werden nach der Konfiguration am Dashboard die eingehenden Daten der angegebenen Agents angezeigt.

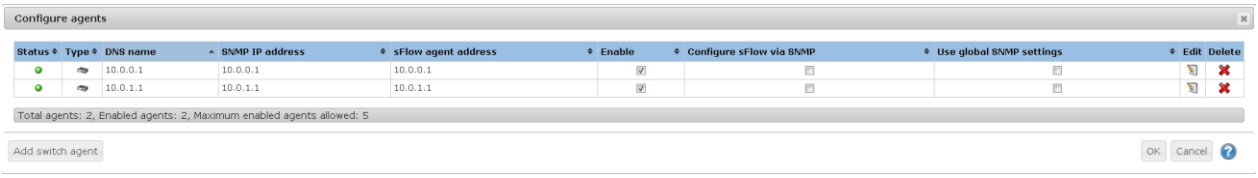


Abbildung 53: Hinzufügen eines Agents, Quelle: Eigene Darstellung.

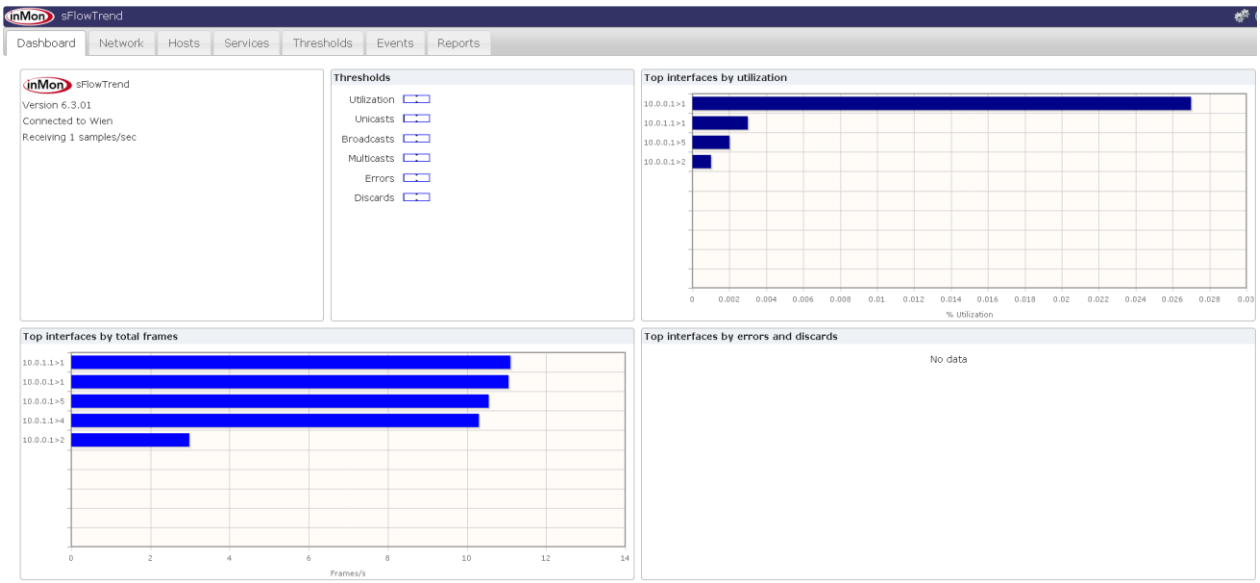


Abbildung 54: Dashboard nach Konfiguration, Quelle: Eigene Darstellung.

7.2.1.3 Grafische Darstellung der Bandbreite

Der Reiter „**Network**“ gibt hauptsächlich eine Übersicht über den tatsächlichen Traffic zu einem Gerät und bildet das „Herz“ der Software. Dabei kann die Anzeige nach verschiedenen Kriterien gefiltert bzw. verändert werden. Das Übersichtsdiagramm zeigt die Daten über den gesamten Traffic eines bestimmten HW-Interfaces einer bestimmten IP an. Je nach Chart-Typ kann die Einheit der Darstellung verändert werden (z.B. Bits/s oder Frames/s). Unter „Root Cause“ werden zusätzlich alle IP-Adressen angezeigt, die als Quell- oder Zieladresse jemals in Verwendung waren und wieviel sie jeweils zum gesamten Traffic beigetragen haben.

7.2.1.4 Filterungsmöglichkeiten

Um eine bestimmte VPN-Verbindung zu überwachen, ist ein Filter nach Source- und Destination-IP notwendig. Dazu klickt man auf das Filtersymbol neben der Switch-Auswahl und ein Textfeld erscheint. In das Textfeld gibt man den gewünschten Filter ein. Folgende Filterungsmöglichkeiten können angewendet werden:

Bezeichnung/IP	Filter
WAN1 (80.120.196.82)	`sourceAddress == "80.120.196.82" || destinationAddress == "80.120.196.82"`
Nur HTTP-Traffic	`sourcePort == "TCP:80" || destinationPort == "TCP:80"`
Nur HTTPS-Traffic	`sourcePort == "TCP:443" || destinationPort == "TCP:443"`
Nur Web-Traffic (HTTP und HTTPS)	`sourcePort == "TCP:80" || destinationPort == "TCP:80" || sourcePort == "TCP:443" || destinationPort == "TCP:443"`

Tabelle 21: Filterungsmöglichkeiten "SFlow Trend".

Bei Anwenden eines Filters wird die Anzeige sofort dementsprechend aktualisiert (siehe Abbildung 55).

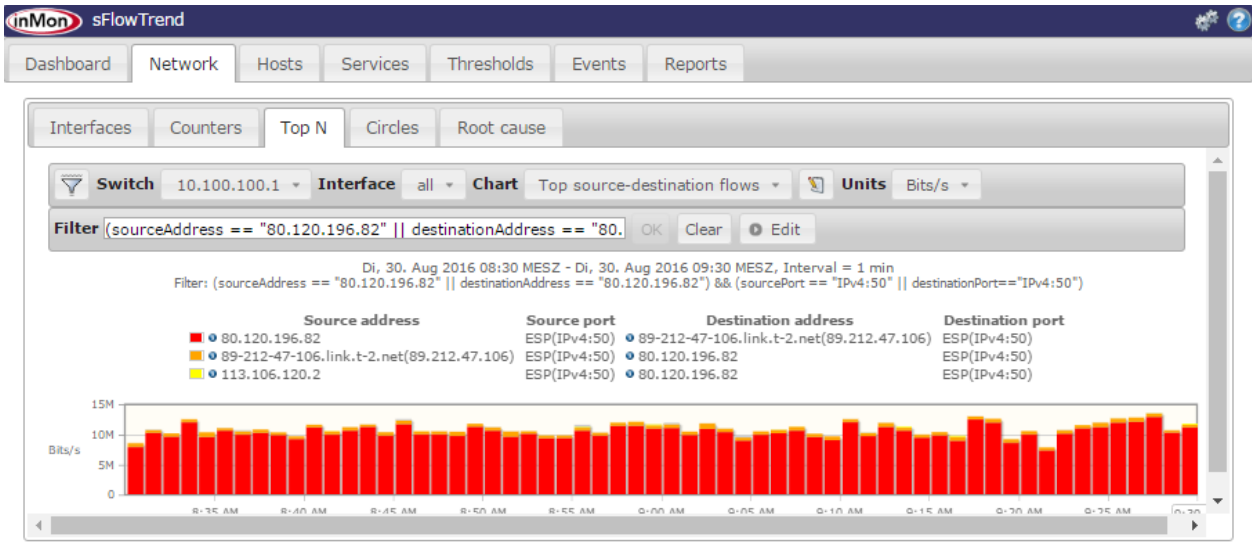


Abbildung 55: Grafische Darstellung einer VPN-Verbindung (durch Filterung), Quelle: Eigene Darstellung.

Als Chart-Option empfiehlt es sich, eine der folgenden Möglichkeiten zu wählen:

- Top Sources: Sortierung nach Menge von Traffic, der von einer bestimmten IP ausgeht
- Top Destinations: Sortierung nach Menge von Traffic, der an eine bestimmte IP gesendet wird
- Top Source-Destination Pairs: Sortierung nach Menge von Traffic, der zwischen zwei bestimmten IP-Adressen ausgetauscht wird
- Top Source-Destination Flows: gleich wie Top Source-Destination Pairs, nur dass auch Source- und Destination-Port in die Wertung miteinbezogen werden
- Top IP-Protocols: Aufschlüsselung nach TCP, ESP und UDP
- Eigene Liste definieren: es ist möglich, eine eigene „Toplist“ zu definieren. Dazu kann man mehrere Felder auswählen. Nützlich sind u.a. „macDestination“, „macSource“, „udpSourcePort“, „tcpSourcePort“, „udpDestinationPort“, „tcpDestinationPort“

Eine Möglichkeit den gesamten Traffic aller Interfaces anzuzeigen, gibt es unter „Counters“ nicht. Dafür kann allerdings ein Workaround angewendet werden (siehe Abbildung 56). Unter „Top N“ kann ein bestimmter Switch ausgewählt werden, dessen Traffic von allen Interfaces man mit der Option „All“ anzeigen kann. Die Chart-Optionen können, wie oben beschrieben, ausgewählt werden.

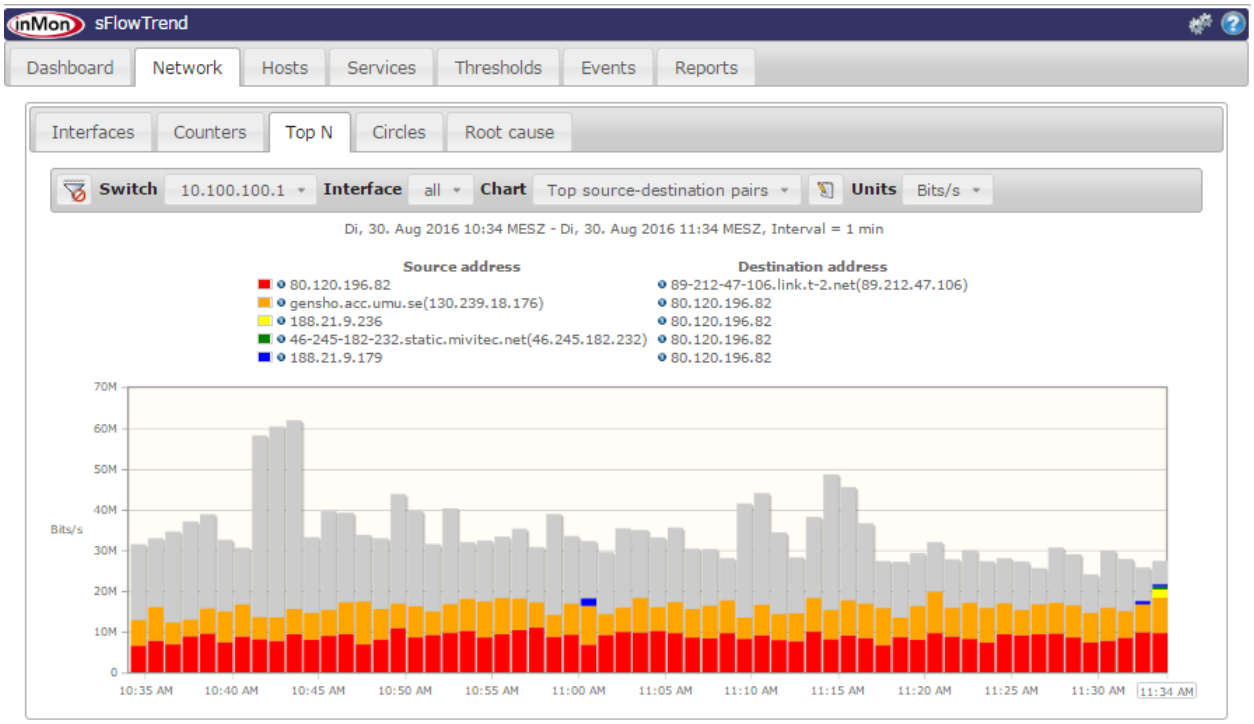


Abbildung 56: Traffic-Darstellung aller Interfaces eines Switchs, Quelle: Eigene Darstellung.

7.2.1.5 Weitere Funktionen

Unter „**Hosts**“ und „**Services**“ findet man keine Anwendung bezüglich VPN Traffic Monitoring. Hier könnte der Agent über SNMP überwacht werden (z.B. Speicherplatz, CPU, Services, etc.). Da diese Thematik allerdings in einem eigenen Server-Monitoring-Tool Anwendung findet, ist der Bereich von „SFlow Trend“ nicht weiter relevant.

Der Bereich „**Thresholds**“ bietet die Möglichkeit, Richtwerte festzulegen. Bei Übersteigen werden diese am Dashboard ersichtlich (Rot = sehr kritisch, Gelb = mittel, Grün = OK).

Der Tab „**Events**“ zeigt jegliche Ereignisse an, die mit dem Server in Verbindung stehen und zu welcher Zeit sie passiert sind. Z.B.: Ein Grenzwert wird erreicht oder jemand verbindet sich zum SFlow-Server. Über das Einstellungsmenü („Configure Events“) kann festgelegt werden was geschehen soll, wenn ein bestimmtes Event passiert (z.B. Senden eines Emails).

Möchte man z.B. die Geschehnisse des Tages zusammenfassen, kann man einen „**Report**“ erstellen. Unter „System Reports“ findet man bereits einige in vorgefertigter Form.

7.2.2 Installation & Konfiguration von „Icinga“

Nun folgt die Beschreibung der ausgewählten Server-Monitoring-Software. Konkret wurde „Icinga2“ (die neueste Version der Software) am Testsystem installiert (erreichbar unter der URL: http://10.0.1.3/icingaweb2/dashboard). Wie bereits in Kapitel 6.3.2.1 erwähnt, wurde Icinga1 als „Fork“[120] von der Monitoring-Software „Nagios“ weiterentwickelt. Die zweite Version von „Icinga“ wurde als

[120] Fork…Abspaltung und Weiterentwicklung einer Software.

eigenständige Software entworfen. Das Monitoring-Tool sieht vor, die Netzwerkumgebung in „Master“ und „Satellites“ aufzuteilen (siehe Abbildung 57). Diese erfüllen folgende Aufgaben:

- **Master**: An die zentrale Überwachungsstelle werden die Daten gesendet und auf der dort lokalisierten Datenbank gespeichert. Auch ein Web-Interface („Icinga Web 2“) kann auf diesem Server für eine bessere Darstellung installiert werden.
- **Satellites**: Stellen die zu überwachenden Geräte dar und senden die Monitoring-Daten an den „Master“.

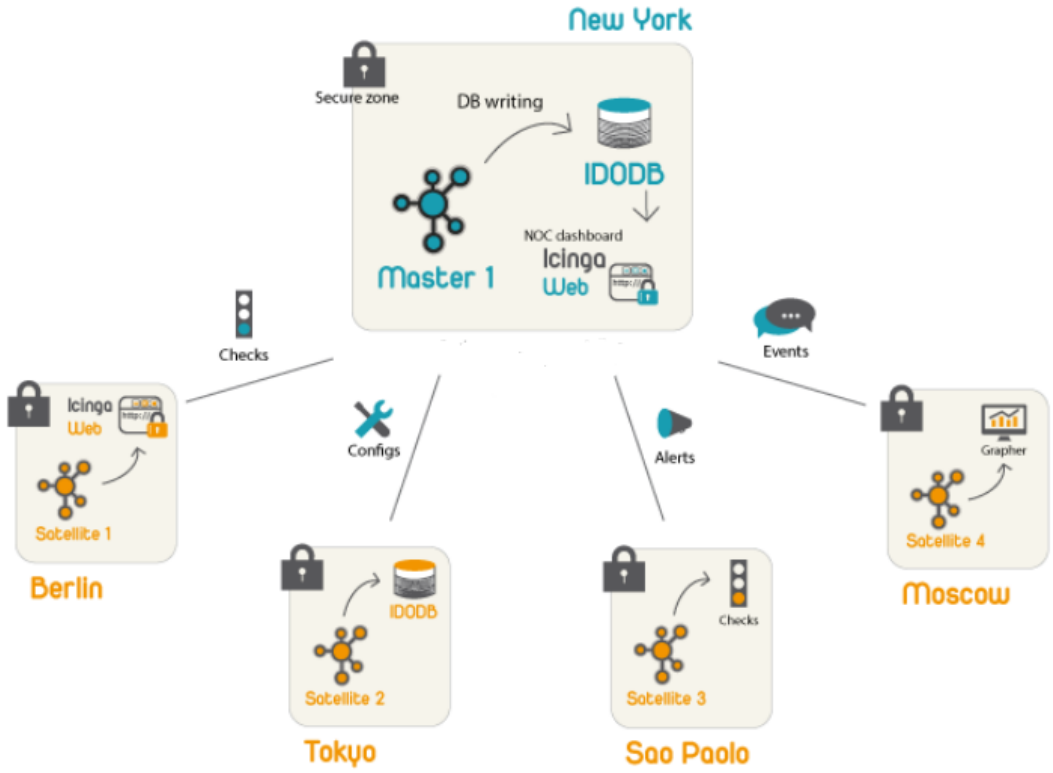


Abbildung 57: Verteiltes Monitoring mit "Icinga", Quelle: The Icinga Project (2016), Online-Quelle [22.September.2016].

7.2.2.1 Installation

Die Installation und Konfiguration ist nur durch Kommandozeilenbefehle bzw. Einträge in Dateien am „Master“ möglich. Das erschwert am Beginn zwar die Einarbeitung, ermöglicht aber nach einer geraumen Einsatzdauer viele Optionen und steigert die Flexibilität. Wie schon „SFlow Trend“, wurde auch „Icinga“ auf einem Debian-Server (Debian GNU/Linux 8 Jessie) installiert.

Zu Beginn der Testphase wurde die Installation des Pakets am „Master“ vorgenommen.

```
apt-get install icinga2
```

Des Weiteren müssen Plugins installiert werden, die es dem Benutzer ermöglichen, einige Monitoring-Services bereits „out-of-the-box“ verwenden zu können.

```
apt-get install nagios-plugins
```

Danach kann die Software bereits am „Master“ gestartet werden. Die dafür benötigte „init“-Datei ist durch den Pfad „`/etc/init.d/icinga2`“ erreichbar und kann mit einem der untenstehenden Befehle verwendet werden:

```
/etc/init.d/icinga2 {start|stop|restart|reload|checkconfig|status}
```

Nach dem bereits gestarteten „daemon“ (Hintergrundprogramm) wird nun die Installation des Web-Interfaces vorgenommen. Zunächst muss dafür eine Datenbank eingerichtet werden. Am Testsystem wurde eine MySQL-Datenbank dafür verwendet. „Icinga“ benötigt zusätzlich ein Modul (DB IDO[121]), welches Konfigurations- und Status-Daten in die Datenbank exportiert:

```
apt-get install mysql-server mysql-client
```

```
apt-get install icinga2-ido-mysql
```

```
mysql -u root -p
```

Die Eingabe des Passworts wird hier erfordert.

```
mysql>  CREATE DATABASE icinga;
```

```
mysql>   GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga';
```

```
mysql> quit
```

Nach der Erstellung der Datenbank, kann das „Icinga2 IDO“-Schema importiert und das Modul aktiviert werden (erfordert einen Neustart von „Icinga“):

```
mysql -u root -p icinga < /usr/share/icinga2-ido-mysql/schema/mysql.sql
```

```
/etc/init.d/icinga2 feature enable ido-mysql
```

```
/etc/init.d/icinga2 restart
```

Neben der Datenbank muss natürlich auch noch ein Web-Server für die Verwendung des Web-Interfaces installiert werden:

```
apt-get install apache2
```

Danach erfolgt die tatsächliche Installation von „Icinga Web 2“:

```
apt-get install icingaweb2
```

Für die Fertigstellung der Installation kann der „Icinga Web 2 Setup Wizard“ unter http://10.0.1.3/icingaweb2/setup verwendet werden. Hier wird man noch aufgefordert, sich mit einem zuvor generierten Token zu identifizieren. Zusätzlich wird noch ein Benutzer angelegt (Benutzername: „icingaweb“), mit dem man auf das Web-Interface zugreifen kann. Danach ist die Installation abgeschlossen. Als Startseite werden bereits überwachte Daten vom „Master“ angezeigt (siehe Abbildung 58).

[121] DB IDO…Database Icinga Data Output.

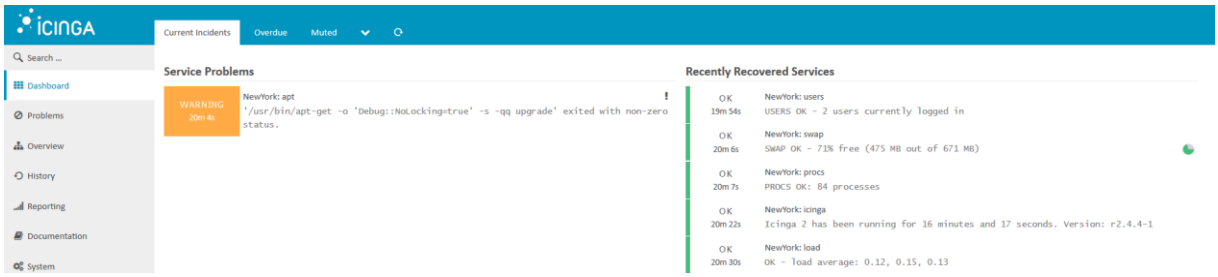


Abbildung 58: Startseite des "Icinga Web 2", Quelle: Eigene Darstellung.

7.2.2.2 Hinzufügen von Hosts/Satellites

Im nächsten Schritt werden sogenannte „Hosts“ oder „Satellites“ hinzugefügt. Dafür muss der Zugriff auf die zu überwachenden Geräte gegeben sein. In der Testphase wurde der Server „Wien“ als „Satellite“ hinzugefügt. Ebenso wie beim „Master“, muss zuvor die Software „Icinga“ auf „Wien“ installiert werden. Danach kann die Konfiguration mithilfe des Befehls „`/etc/icinga2 node wizard`“ begonnen werden (siehe Abbildung 59).

Dabei müssen in erster Linie der Name und die IP des „Masters“ angegeben werden, um die Verbindung herzustellen. Als Authentifizierung wird hier ein sogenanntes „Ticket“ verwendet, welches am „Master“ generiert wird und bei der „Satellite“-Konfiguration angegeben werden muss.

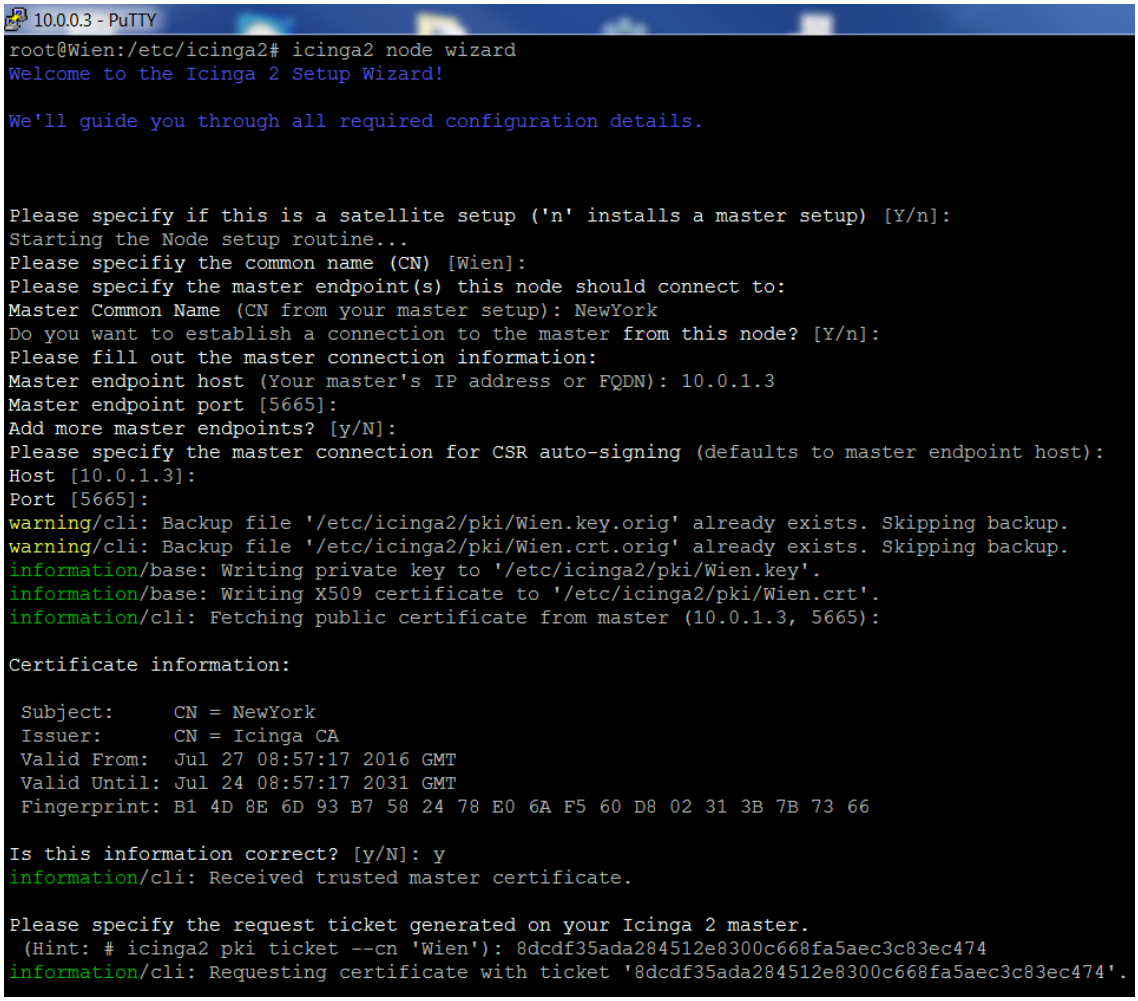


Abbildung 59: Hinzufügen eines "Satellites", Quelle: Eigene Darstellung.

Nach erfolgreicher Installation und Konfiguration des „Satellites“ wird dieser, wie in Abbildung 60 ersichtlich, im „Icingaweb“ angezeigt.

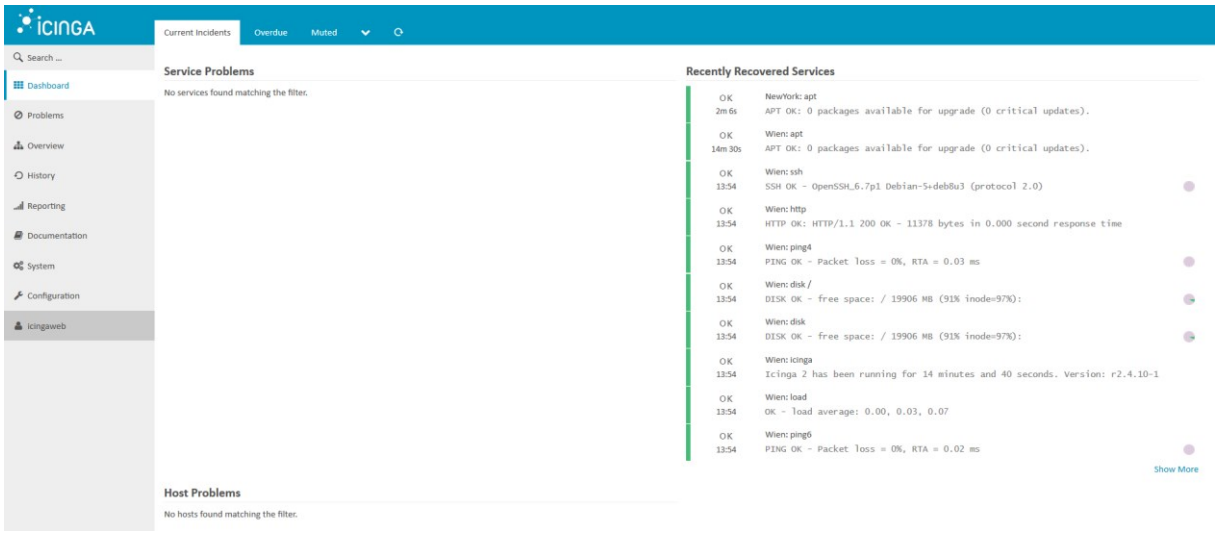


Abbildung 60: Dashboard nach Hinzufügen eines „Satellites“, Quelle: Eigene Darstellung.

7.2.2.3 Services

Für die Evaluierung am Testsystem werden folgende Parameter am Test-Server „Wien“ überwacht.

- APT: Prüft, ob Updates auf dem Agent verfügbar sind
- DISK: Prüft den freien/belegten Speicherplatz auf einem Laufwerk (Grenzwerte können eingestellt werden)
- HOSTALIVE (PING): Prüft, ob der Agent/Host erreichbar ist.
- HTTP: Prüft, ob der HTTP-Service am Agent ordnungsgemäß funktioniert. (auch https möglich)
- LOAD: Auslastung des Systems
- PROCS: Prüft die Anzahl der Prozesse, die am Agent ausgeführt werden. (Meldet WARNING oder CRITICAL bei Überschreitung der Grenzen)
- SWAP: Prüft den freien SWAP-Speicherplatz[122]
- USERS: Wie viele User gerade am Agent eingeloggt sind

Sogenannte „Services“ ermöglichen das Monitoring, welche in der „services.conf“-Datei (unter: „`/etc/icinga2/conf.d`“) definiert werden können. Dabei wird jede Definition folgendermaßen aufgebaut:

- „apply“: Name des Service
- „import“: Der neu erstellte Service kann von einem bereits bestehenden Template abgeleitet werden

[122] SWAP-Speicherplatz…reservierter Speicher auf der Festplatte der zum Einsatz kommt, wenn der RAM-Speicher bereits voll ist.

- „check_command“: Der tatsächlich auszuführende Befehl
- „assign where“: Festlegung, für welche Hosts der Service angewendet werden soll

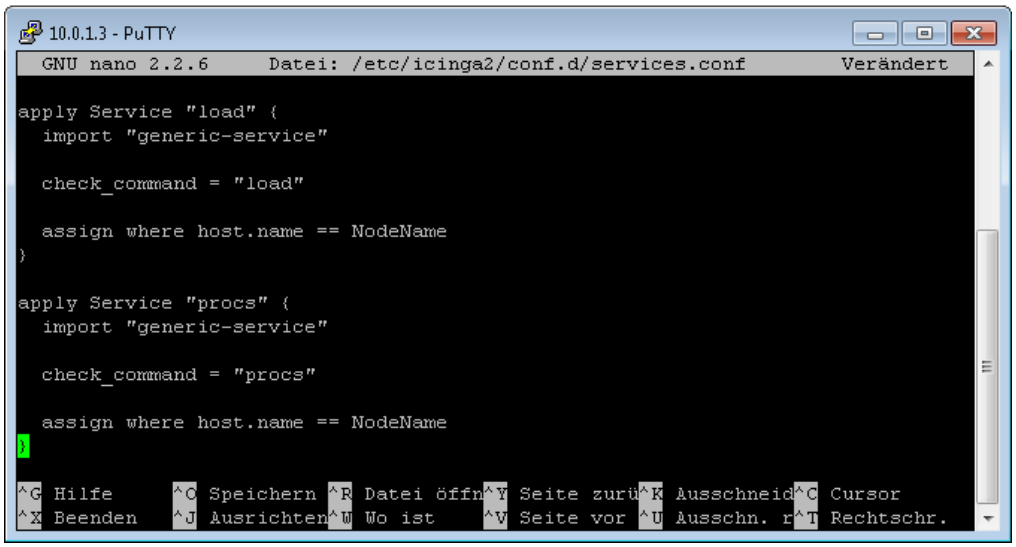


Abbildung 61: Definition von Services, Quelle: Eigene Darstellung.

7.2.2.4 „agentless“-Monitoring

Eine weitere Möglichkeit zur Überwachung eines Servers, bietet das „agentless“-Monitoring. Wie der Name schon vermuten lässt, ist hier kein Zugriff auf das zu überwachende Gerät notwendig. Dies lässt sich allein durch Einstellungen am „Master“ durchführen. Einen Vorteil bringt diese Monitoring-Variante, wenn der Server nicht zugänglich ist, zu wenig Speicherplatz für die Installation einer Software zur Verfügung steht, oder der Host nur wenig Möglichkeiten an zu überwachenden Parametern bietet, welche sich auch über „agentless“-Monitoring verfolgen lassen.

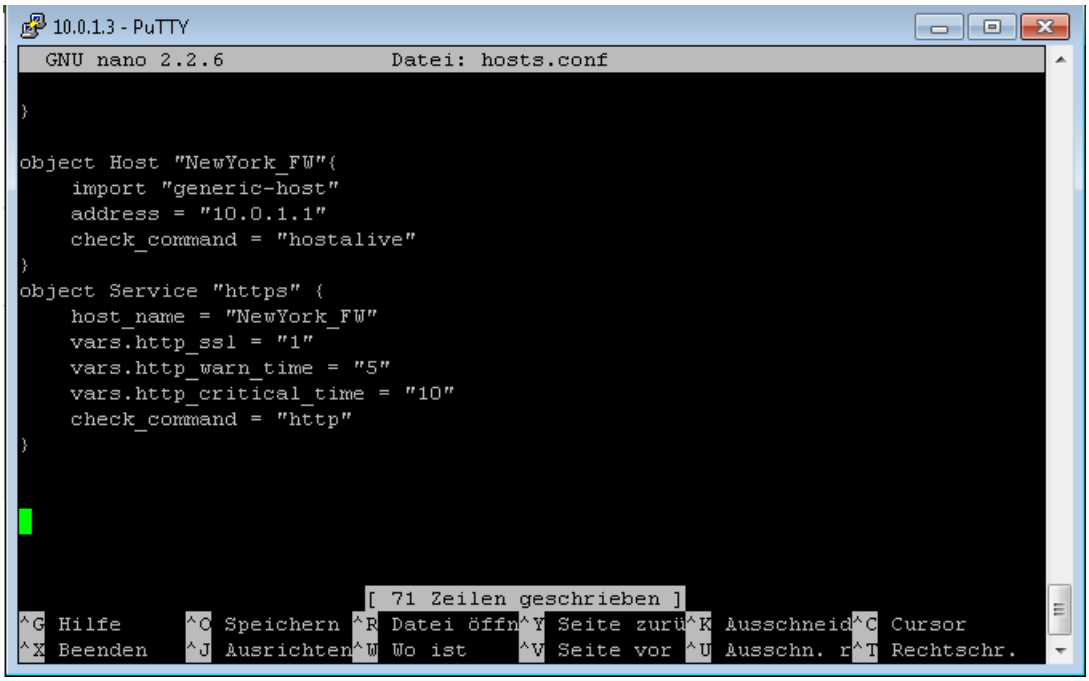


Abbildung 62: Konfiguration für "agentless"-Monitoring, Quelle: Eigene Darstellung.

Die Konfiguration erfolgt in der „hosts.conf“-Datei (siehe Abbildung 62), welche sich unter „`/etc/icinga2/conf.d`“ befindet. Neben dem Hinzufügen des Hosts „NewYork_FW“ (FW…Firewall) werden hier auch die auszuführenden Services definiert. In diesem konkreten Fall werden ein „Hostalive“ und ein HTTPS-Check durchgeführt.

Mit dem Befehl „`icinga2 daemon -C`“ wird die Konfigurationsdatei noch einmal überprüft, bevor das Programm neu geladen („`reload`“) und gestartet („`restart`“) werden muss. Danach ist am Dashboard von „Icingaweb2“ der neue Host bereits ersichtlich (siehe Abbildung 63).

Abbildung 63: Anzeige der "agentless"-Checks am Dashboard, Quelle: Eigene Darstellung.

7.2.2.5 Gruppen

Um bei einer großen Anzahl an überwachten Geräten den Überblick zu bewahren, stellt „Icinga“ die Funktion einer Gruppe zur Verfügung. Dabei können beliebig viele Gruppen erstellt und jeweilige Hosts hinzugefügt werden. Im Falle der Firma LOGICDATA bietet sich die Erstellung der Gruppe „Lohnfertiger-Server-Gruppe“ an, welche in die Konfigurationsdatei „groups.conf“ (Pfad: „`/etc/icinga2/conf.d`“) eingetragen wird (siehe Abbildung 64). Zusätzlich wird ein Host-Template erstellt, welches in den Einstellungen beim Host angegeben werden muss.

```
object HostGroup "Lohnfertiger-Server-Gruppe" {
  display_name = "Lohnfertiger-Server-Gruppe"
}

template Host "Lohnfertiger-Server" {
  groups += [ "Lohnfertiger-Server-Gruppe" ]
}
```

```
object Host "Wien" {
        import "Lohnfertiger-Server"
        check_command = "cluster-zone"
}
```

Abbildung 64: Erstellung einer Gruppe und Hinzufügen eines Hosts, Quelle: Eigene Darstellung.

Im „Icingaweb“ wird nun die Gruppe mit den dazugehörigen Hosts unter „Overview“, „Hostgroups“ angezeigt.

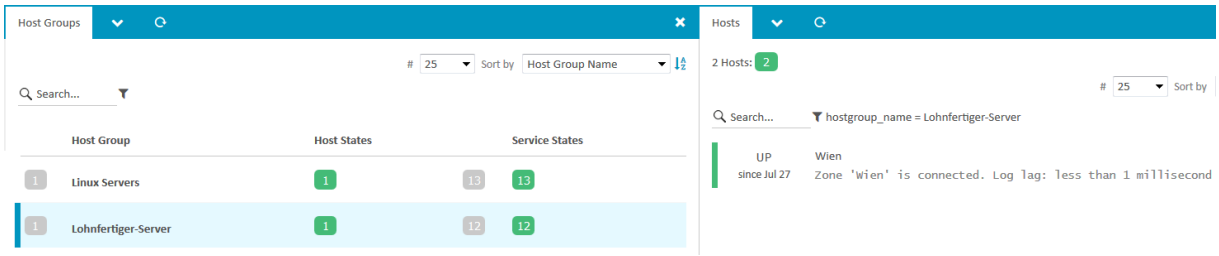


Abbildung 65: Anzeige der Gruppe im "Icingaweb", Quelle: Eigene Darstellung.

7.2.2.6 Notifications

Um über Monitoring-Ereignisse bestmöglich informiert zu sein, können Mitteilungen an festgelegte Personen gesendet werden. Im Wesentlichen ähnelt dieser Dienst im Aufbau einem Service (wie bereits

oben beschrieben). Sogenannte „Notifications“ werden entweder auf Services oder auf Hosts angewendet. Im Falle der Services werden nur dann Mitteilungen versendet, wenn die Variable „notification.mail“ definiert ist. Die Konfiguration dieser Variante könnte wie in Abbildung 66 ersichtlich aufgebaut sein.

Abbildung 66: Definition von "Notifications" auf Services, Quelle: Eigene Darstellung.

Bei Anwendung auf Hosts muss nur das Schlüsselwort „Service“ in der Definition ausgetauscht werden. Zusätzlich ist noch zu erwähnen, dass für das Versenden von Nachrichten das „Notification-Feature“ aktiviert sein muss. Dies kann mithilfe des Befehls „`icinga2 feature list`“ überprüft werden (siehe Abbildung 67).

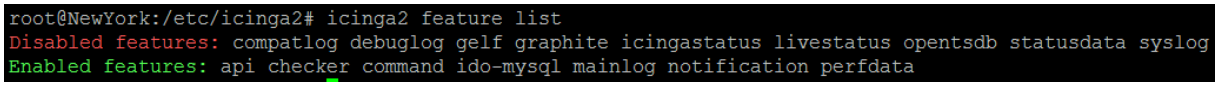


Abbildung 67: Überprüfung der Features, Quelle: Eigene Darstellung.

8 ZUSAMMENFASSUNG UND AUSBLICK

Nach den Analysen der Protokolle im Theorieteil und der Monitoring-Lösungen im Praxisteil dieser Arbeit, widmet sich nun das letzte Kapitel der Zusammenfassung und dem Ausblick.

8.1 Zusammenfassung & Empfehlungen

Zusammenfassend ist zu sagen, dass sich besonders die Wichtigkeit eines sicheren Remote-Monitorings im Zuge der Arbeit herauskristallisierte. Da die wirtschaftlichen Folgen eines Ausfalls nicht unerheblich sind, sollte man sich bereits zuvor intensiv diesem Thema widmen, um mögliche Zwischenfälle zumindest frühzeitig zu erkennen, bestenfalls aber verhindern zu können. Aus diesem Grund können folgende Empfehlungen an LOGICDATA weitergegeben werden:

1. Erstellung eines VPN-Threat-Modells: Mögliche Angriffsszenarien müssen erkannt und Gegenmaßnahmen definiert werden. Um bestmögliche Sicherheit gewährleisten zu können, empfiehlt es sich, dieses regelmäßig zu aktualisieren und auf mögliche Änderungen zu reagieren.
2. Prüfung des „TINA“-Protokolls in Zukunft: Da eine sofortige Umstellung der verwendeten Hardware zu kostenintensiv wäre, ist die Umstellung auf das „TINA“-Protokoll zum heutigen Zeitpunkt nicht möglich. Sollte es jedoch in Zukunft zu einem Hardware-Tausch kommen, sollte der Einsatz des Protokolls noch einmal eingehend geprüft werden, da es doch Vorteile, wie z.B. Traffic Intelligence (Auswählen alternativer Kommunikationswege), Traffic Shaping (optimale Nutzung der Bandbreite) oder das Auswählen verschiedener Transport-Modi, was die Performance und die Stabilität der Verbindung je nach Gegebenheit erhöht, bietet.
3. Einsatz der ausgewählten Monitoring-Lösungen

8.2 Fazit & Ausblick

Für den Autor besonders interessant waren die Analyse der „IoT“-Protokolle und das Prüfen einer Anwendung bei LOGICDATA, da diese grundsätzlich nicht für eine solche Art der Verbindung ausgelegt sind, aber theoretisch durchaus für einen Einsatz infrage kommen würden (z.B. das MQTT-Protokoll). Eine weitere essentielle Erkenntnis aus dieser Arbeit ist, wie bereits in der Zusammenfassung erwähnt, der Nutzen und die Wichtigkeit eines Monitorings überhaupt, um die bestmögliche Verfügbarkeit gewährleisten zu können.

Als nächsten Schritt können nach dem Evaluierungsprozess der Monitoring-Lösungen und den Versuchen am Testaufbau, wie bereits in den Empfehlungen erwähnt, die ausgewählten Tools („SFlowTrend“ und „Icinga2“) auf den tatsächlich verwendeten Lohnfertiger-Servern installiert und getestet werden. Mittelfristig wird das derzeit eingesetzte Überwachungs-Tool „Hydra“ durch die neuen Lösungen ersetzt. Die neuen Überwachungsmöglichkeiten werden in Zukunft das Risiko eines Ausfalls bzw. auch die finanziellen Auswirkungen für LOGICDATA minimieren.

Eine Vertiefung dieser Arbeit könnte sowohl im Bereich der Sicherheit, also die konkrete Ausarbeitung eines VPN-Threat-Modells für LOGICDATA, erfolgen, als auch in der praktischen Analyse und Tests der Ergebnisse aus Kapitel 4.5 „Empfehlung einer Alternative“ zu einem späteren Zeitpunkt. Eine langfristige

Untersuchung der VPN-Technologien im Allgemeinen (Wie werden VPN-Lösungen in Zukunft realisiert? Wie wird deren Sicherheit gewährleistet werden?) könnte ebenfalls sehr von Nutzen für LOGICDATA sein, um schon frühzeitig auf eine eventuelle Technologieänderung reagieren zu können.

LITERATURVERZEICHNIS

Gedruckte Werke (21)

Aebi, Daniel (2004): *Praxishandbuch Sicherer IT-Betrieb, Risiken erkennen, Schwachstellen beseitigen, IT-Infrastrukturen schützen*, 1. Auflage, Betriebswirtschaftlicher Verlag Dr. Th. Gabler/GWV Fachverlage GmbH, Wiesbaden

Beasley, Jeffrey; Nilkaew, Piyasat (2012): *Networking Essentials*, 3. Auflage, Pearson Education Inc., New Jersey

Doraswamy, Naganad; Harkins, Dan (2003): *IPSec, The new security standard for the internet, intranets, and virutal private networks*, 2. Auflage, Prentice Hall PTR, New Jersey

Guichard, Jim; Pepelnjak, Ivan (2001): *MPLS and VPN Architectures*, Cisco Press, Indianapolis

Hein, Mathias (2000): *TCP/IP, Internet-Protokolle im professionellen Einsatz*, 5. Auflage, MITP-Verlag GmbH, Bonn

Henmi, Anne; Lucas, Mark; Singh, Abhishek; Cantrell, Chris (2006): *Firewall Policies and VPN Configuration*, 1. Auflage, Syngress Publishing Inc., Rockland

Hiles, Andrew (2002): *The Complete Guide to IT Service Level Agreements, Aligning IT Service to Business Needs*, 3. Auflage, The Rothstein Catalog On Service Level Books Rothstein Associates Inc., Conneticut

Kohne, Andreas; Ringleb, Sonja; Yücel, Cengizhan (2015): *Bring your own Device, Einsatz von privaten Endgeräten im beruflichen Umfeld - Chance, Risiken und Möglichkeiten*, 1. Auflage, Springer Fachmedien, Wiesbaden

Lipp, Manfred (2001): *VPN - Virtuelle Private Netzwerke*, 1. Auflage, Addison-Wesley Verlag, München

Mann, H.; Bernhard, M.G.; Lewandowski, W. (2004): *Service-Level-Management in der IT, Wie man erfolgskritische Leistungen definiert und steuert*, 5. Auflage, Symposion Publishing GmbH, Düsseldorf

Mauro, Douglas; Schmidt, Kevin (2005): *Essential SNMP*, 2. Auflage, O'Reilly Media Inc., Sebastopol

Mel, H.X.; Baker, Doris (2001): *Cryptography Decrypted*, 1. Auflage, Addison Wesley, Canada

Pasley, Keith (2002): *VPN Deployment and Evaluation Strategy*, in: Tipton, Harold; Krause, Micki (Hrsg.): *Information Security Management, Handbook*, 4. Auflage, Auerbach Publications, Florida, S. 149-177

Reiser, Christian (2000): *Internet - die Sicherheitsfragen*, 2. Auflage, Wirtschaftsverlag Carl Ueberreuter, Wien/Frankfurt

Schäfer, Günter; Roßberg, Michael (2014): *Netzsicherheit*, 2. Auflage, dpunkt.verlag, Heidelberg

Schrey, J.; Bernhard, M.G.; Mann, H.; Lewandowski, W. (2006): *Praxishandbuch Service-Level-Management, Die IT als Dienstleistung organisieren*, 2. Auflage, Symposion Publishing GmbH, Düsseldorf

Schudel, Gregg; Smith, David (2008): *Router Security Strategies, Securing IP Network Traffic Planes*, 1. Auflage, Cisco Systems Inc., Indianapolis

Scott, Charlie; Wolfe, Paul; Erwin, Mike (1999): *Virutal Private Networks*, 2. Auflage, O'Reilly Associates, Inc., Sebastopol

Shea, Richard (2000): *L2TP - Implementation and Operation*, 1. Auflage, Addison Wesley Longman, Inc., Massachusetts

Spenneberg, Ralf (2010): *VPN mit Linux, Grundlagen und Anwendung virtueller privater Netzwerke mit Open-Source-Tools*, 2. Auflage, Addison-Wesley Verlag, München

Stark, Shubhangi (2001): *Sichere IT-Kommunikation über unsichere Netze*, 1. Auflage, Diplomica GmbH, Hamburg

Thomas, Stephen (2000): *SSL and TLS Essentials, Securing the Web*, 1. Auflage, John Wiley & Sons, Inc., Canada

Tiller, James (2002): *The (In) Security of Virtual Private Networks*, in: Tipton, Harold; Krause, Micki (Hrsg.): *Information Security Management, Handbook*, 4. Auflage, Auerbach Publications, Florida, S. 215-271

Tiller, James (2002): *Leveraging Virtual Private Networks*, in: Tipton, Harold; Krause, Micki (Hrsg.): *Information Security Management, Handbook*, 4. Auflage, Auerbach Publications, Florida, S. 273-302

Vishwarkama, Alok (2016): *Virtual Private Networks*, in: Dileep, Kumar; Manoj, Kumar; Jayanthi, M.K. (Hrsg.): *Network Security Attacks and Countermeasures*, 1. Auflage, IGI Global, Hershey, S. 78-114

Online-Quellen (26)

Cisco Systems, Inc. (2009): *http://www.cisco.com*
http://www.cisco.com/c/en/us/td/docs/security/security_management/cisco_security_manager/security_manager/4-1/user/guide/CSMUserGuide_wrapper/vpchap.pdf [Stand: 11.Juli.2016]

techtarget.com (2006): *searchsecurity.techtarget.com*
http://searchsecurity.techtarget.com/definition/threat-modeling [Stand: 29.Juli.2016]

MQTT.org (2016): *www.mqtt.org*
http://mqtt.org/faq [Stand: 30.Juli.2016]

Zoho Corp. (2016): *https://www.manageengine.com*
https://www.manageengine.com/network-monitoring/what-is-snmp.html#snmp-basic-commands [Stand: 8.August.2016]

inMon Corp. (2007): *http://www.sflow.org*
http://www.sflow.org/process/process6.htm [Stand: 11.August.2016]

Barracuda Networks (2015): *https://campus.barracuda.com*
https://campus.barracuda.com/product/nextgenfirewallf/article/NGF62/VPNSiteToSiteTINA/ [Stand: 12.August.2016]

MILS Electronic (2010): *http://www.mils.com*
http://www.mils.com/uploads/tx_rfknowledge/KB_MilsVPN_TINA_Protocol_mils_electronic_100714.pdf [Stand: 12.August.2016]

ClickSSL (2013): *https://www.clickssl.net*
https://www.clickssl.net/blog/vpn-security-flaws-and-its-prevention [Stand: 14.September.2016]

The Government of the Hong Kong Special Administrative Region (2008): *http://www.infosec.gov.hk*
http://www.infosec.gov.hk/english/technical/files/vpn.pdf [Stand: 14.September.2016]

The Icinga Project (2016): *https://www.icinga.org*
https://www.icinga.org/products/icinga-2/distributed-monitoring/ [Stand: 22.September.2016]

Behringer, Michael; Morrow, Monique (2005): *www.ciscopress.com*
http://www.ciscopress.com/articles/article.asp?p=418656 [Stand: 29.Juli.2016]

Chen, Xi (2014): *http://www.cse.wustl.edu*
http://www.cse.wustl.edu/~jain/cse574-14/ftp/coap/#sec2-1 [Stand: 1.August.2016]

Elektronik Kompendium (2016): *http://www.elektronik-kompendium.de*
http://www.elektronik-kompendium.de/sites/kom/0301201.htm [Stand: 12.Juli.2016]

Elektronik Kompendium (2016): *http://www.elektronik-kompendium.de/*
http://www.elektronik-kompendium.de/sites/net/0906141.htm [Stand: 15.Juli.2016]

Elektronik Kompendium (2016): *http://www.elektronik-kompendium.de*
http://www.elektronik-kompendium.de/sites/net/1410151.htm [Stand: 26.Juli.2016]

Frahim, Jazib; Huang, Qiang (2008): *http://www.ciscopress.com*
http://www.ciscopress.com/articles/article.asp?p=1218144&seqNum=3 [Stand: 14.September.2016]

Gupta, Rahul (2014): *https://www.ibm.com*
https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?lang=en [Stand: 22.August.2016]

Hekerens, Chrisitan (2001): *Johannes Kepler Universität*
http://www.fim.uni-linz.ac.at/Diplomarbeiten/diplomarbeit_hekerens/index.htm [Stand: 1.Juli.2016]

Jaffey, Toby (2014): *www.eclipse.org*
http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php [Stand: 30.Juli.2016]

Jasinska, Elisa (2004): *http://www.sflow.org*
http://www.sflow.org/developers/diagrams/sFlowV5Datagram.pdf [Stand: 11.August.2016]

Patterson, Michael (2010): *https://www.plixer.com/*
https://www.plixer.com/blog/netflow/bandwidth-monitoring-snmp-vs-netflow/ [Stand: 5.September.2016]

Phaal, Peter; Lavine, Marc (2004): *http://www.sflow.org/*
http://www.sflow.org/sflow_version_5.txt [Stand: 11.August.2016]

Remus, Michael (2015): *https://anonymweb.de*
https://anonymweb.de/vpn-protokoll-vergleich-pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/ [Stand: 26.Juli.2016]

Schneider, Stan (2013): *http://electronicdesign.com*
http://electronicdesign.com/iot/understanding-protocols-behind-internet-things [Stand: 22.August.2016]

Stanford-Clark, Andy; Truong, Hong (2013): *http://mqtt.org*
http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf [Stand: 14.September.2016]

Stansberry, James (2015): *http://electronicdesign.com*
http://electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot [Stand: 22.August.2016]

Normen (3)

Case, J; Mundy, R; Partain, D.; Stewart, B. (Hrsg.) (1999): *Request for Comments: 2570: Introduction to Version 3 of the Internet-standard Network Management Framework*

Claise, Benoit (Hrsg.) (2004): *Request for Comments: 3954: Cisco Systems NetFlow Services Export Version 9*

Shelby, Z.; Hartke, K.; Bormann, C. (Hrsg.) (2014): *Request for Comments: 7252: The Constrained Application Protocol (CoAP)*

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS