HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
INFORMATIK

SWITCH
Serving Swiss Universities

Integration des HAPviewers in das NfSen Framework

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2010/2011

Autoren:	Reto Schneider, Sebastian Hügli
Betreuer:	Prof. Eduard Glatz
Projektpartner:	SWITCH, Zürich, Peter Haag

Erklärung über Eigenständigkeit der Arbeit

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Rapperswill, December 23, 2010:

Name, Unterschrift

Name, Unterschrift

Abstract

Aufgabenstellung

Es bestehen zwei eigenständige Tools, die Webapplikation NfSen, welche Netflow Daten sammelt und anzeigt, sowie die Fat Client Applikation HAPviewer, die es erlaubt, grosse Mengen von Netzwerkdaten zu visualisieren.
Die Aufgabe dieser Arbeit besteht in der Integration der Visualisierungsfunktionalität des HAPviewers in NfSen.

Ziel der Arbeit

Die Funktionalität der Standalone Applikation HAPviewer soll mit Hilfe eines Plugins in die NfSen Umgebung integriert werden. Das Plugin sollte so in die bestehende Applikation eingebaut werden, das es einfach zu erreichen und bedienen ist.
Die erstellte Software sollte die Visualisierung aller Netzwerkverbindungen für eine vom Benutzer in NfSen ausgewählte IP Adresse in Form eines HAP Graphlets ermöglichen. Aus dieser Graphik sollte es möglich sein, per Mausklick zu weiteren Detailinformationen zu gelangen.
Dem NfSen Benutzer soll mit dem Plugin eine zusätzliche Möglichkeit gegeben werden, die gesammelten Netflow Daten zu visualisieren und analysieren.

Lösung

NfSen wurde um Einsprungspunkte für das HAP4NfSen Plugin erweitert. Über diese können die Aktivitäten einer IP Adresse per Mausklick als Graph visualisiert werden.
Das Plugin besteht aus zwei Teilen: Ein Frontend, welches in PHP geschrieben ist und die gesammte Darstellung übernimmt, sowie einem in Perl realisierten Backend, welches durch Aufrufe von NfDump, der modifizierten HAP-Library sowie GraphViz Daten filtert und Graphiken generiert.
Die dargestellten HAP Graphlets verwenden das SVG Format, welches erlaubt, innerhalb der Graphik zu zoomen oder diese zu verschieben. Zudem sind alle Knoten des Graphlets klickbar, was eine Drill-Down Funtionalität ermöglicht. Weitere Möglichkeiten wie eine Undo Funktion oder Hilfe vereinfachen die Bedienung.

Contents

Part I
Technischer Bericht

1 Ergebnissbericht

1.1 Domain Modell

1.1.1 NfSen

NfSen ist unterteilt in ein Frontend, welches für die Behandlung von Benutzeranfragen zuständig ist, sowie ein Backend, welches Daten für das Frontend beschafft und periodische Tasks ausführt. Um auf Netflow Daten zuzugreifen, wird NfDump verwendet.
Sowohl Front- als auch Backend lassen durch Plugins um zusätzliche Funktionen erweitern.

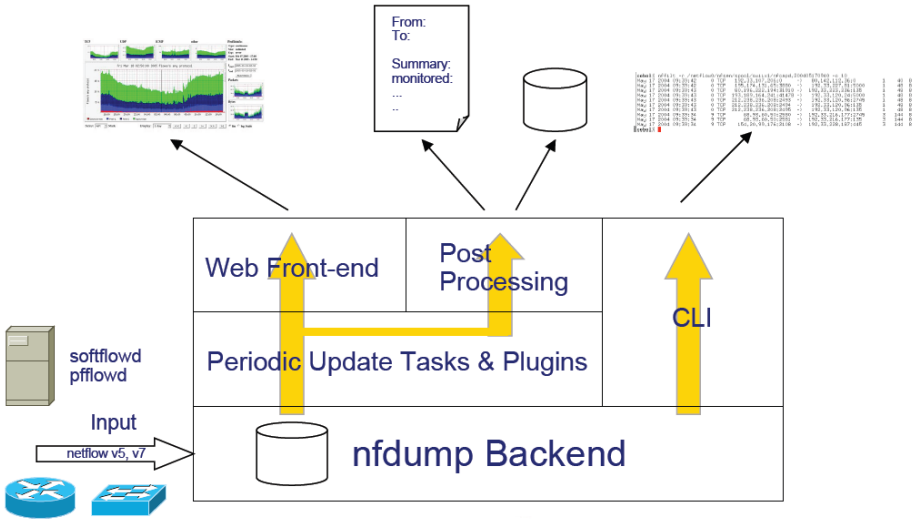

Figure 1: Übersicht zu NfSen[4]

1.1.2 NfDump

Jede in NfSen konfigurierte Datenquelle besitzt ihren eigenen nfcapd Listener, der die gesendeten Netflow Daten in Empfang nimmt. Die empfangenen Daten werden in einzelne Dateien geschrieben. Sender solcher Netflow Daten sind im Normalfall Router. NfDump ist in der Lage, Auswertungen über eine oder mehrere dieser Dateien zu erstellen und das Resultat in Tabellenform auszugeben. NfDump kann über die Kommandozeile aufgerufen werden. In NfSen wird die Anwendung vom Backend aufgerufen, um auf Netflow Daten zuzugreifen.

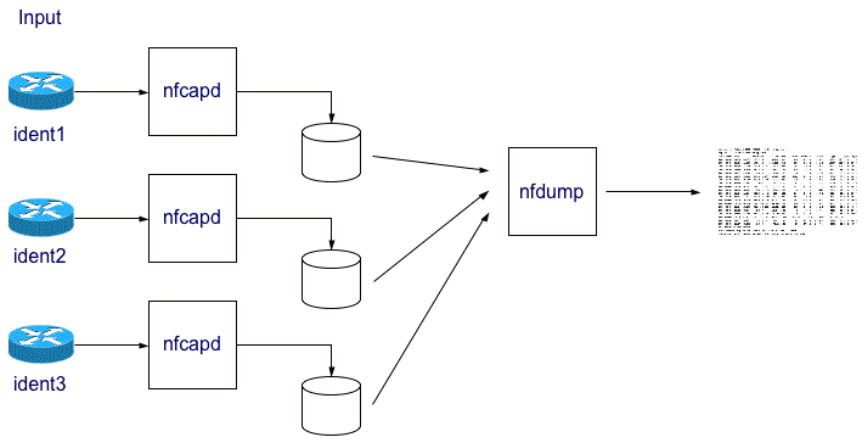

Figure 2: Übersicht zu NfDump[3]

1.2 Requirements

1.2.1 Anwendungsspezifikation

Nr.	Anforderung
A01	Aufruf der HAP-Grafik aus NfSen heraus muss möglich sein
A02	Nach Aufruf der HAP-Grafik soll von dieser aus in geeigneter Form auf weiterführende HAP-Grafiken zugegriffen werden können

1.2.2 Nichtfunktionale Anforderungen

Leistung

Das Plugin sollte in der Lage sein, grosse NetFlow Dateien(bis zu 250MB) innerhalb von Sekunden zu verarbeiten.

Bedienbarkeit

Die Software muss von erfahrenen Benutzern ohne Schulung oder Einarbeitung bedienbar sein.
Die Bedienung soll weitgehend selbsterklärend sein. Bei Unsicherheit soll der Benutzer durch eingebaute Hilfeinformationen unterstützt werden.

Skalierbarkeit

Das Plugin soll auf kleineren wie auch grösseren Systemen mit NfSen installationen verwendet werden können.

1.2.3 Use Cases

Essential

Name:	HAP Graphlet anzeigen
Description:	Ein Benutzer ruft zu einer bestimmten IP aus einer Liste von Netflows(NfDump Ausgabe innerhalb von NfSen) ein HAP Graphlet auf.
Preconditions:	Der Benutzer hat sich innerhalb von NfSen eine NfDump Liste mit Daten nach Wahl anzeigen lassen. Die Liste enthält mindestens eine IP Adresse.
Postconditions:	Ein Fenster mit einem HAP Graphlet wird dem Benutzer angezeigt.

Ablauf:

1. Der Benutzer möchte ein HAP Graphlet zu einer bestimmten IP innerhalb einer Liste von Netflows sehen.
2. Der Benutzer wählt die gewünschte IP Adresse.
3. Das System zeigt dem Benutzer das erstellte HAP Graphlet in einem neuen Fenster an.
4. Ende des Use Cases.

Fully Dressed

Name: HAP Graphlet Drilldown

Scope:	System under Design
Level:	User Goal
Primary Actor:	Benutzer der Applikation

Stakeholders and Interrests:

- Benutzer
  - Möchte mehr Details zu einem bestimmten Teil des HAP Graphlets sehen.

Preconditions:

- Benutzer befindet sich auf der HAPViewer Plugin Seite in NfSen.
- Dem Benutzer wird bereits ein HAP Graphlet angezeigt.
- Das angezeigte Graphlet besitzt mindestens einen zusammengefassten Knoten.

Success Guarantee: Ein neues HAP Graphlet wird dem Benutzer angezeigt.

Main Success Scenatio:

1. Der Benutzer möchte Details zu einem zusammengefassten Knoten eines angezeigten HAP Graphlet sehen.
2. Der Benutzer wählt den gewünschten Knoten aus.
3. Das System hebt die gewünschte Aggregation auf und zeigt dem Benutzer ein neues HAP Graphlet an. Alle Knoten, welche zuvor nicht an der Aggregation beteiligt waren, werden nicht mehr angezeigt.
4. Ende des Use Cases.

Extentions:

- Wenn bei der Erstellung des neuen Graphlets ein technischer Fehler auftritt, wird der Benutzer mit einer Meldung darüber informiert.
- Wenn der Benutzer anstatt eines summarisierten einen gewöhnlichen Knoten auswählt, so werden beim nächsten Graphlet nur noch die mit dem ausgewählten Knoten verbundenen Knoten angezeigt.

Technology and Data Variations List:	Die verwendeten Rohaten werden von NfSen zur Verfügung gestellt.
Frequency of Occurrence:	Pro Benuter: Mehrmals pro Minute

Diagram

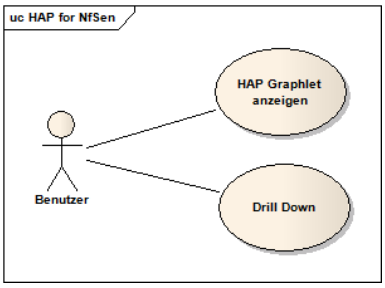

Figure 3: Use Case Übersicht

1.3 Schnittstellen

Die folgenden Abschnitte beschreiben alle wichtigen Schnittstellen. Aufrufe unserer eigenen Module werden dabei nicht beschrieben.

1.3.1 NfSen - HAP4NfSen Plugin

Bis auf eine kleine Ausnahme, handelt es sich bei HAP4NfSen um ein gewöhnliches Plugin für NfSen. Die Plugin Schnittstelle ist ausführlich im NfSen Plugin Writers Guide[5] beschrieben.

Einstiegspunkt

Um es Benutzern zu erlauben, per Mausklick aus NfSen zum HAP4NfSen Plugin zu wechseln, wurden entsprechende Links in die NfSen "Details" Seite integriert.

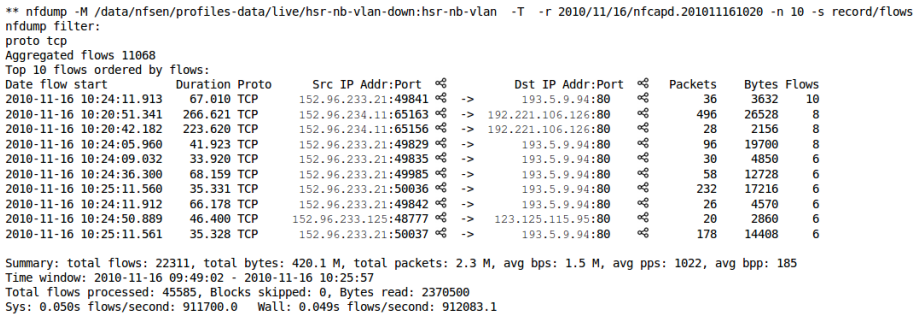

Figure 4: Einstiegspunkt für das Hap4NfSen Plugin in NfSen

Um dies zu ermöglichen war es nötig, die dazugehörige Seite(details.php) anzupassen. Dabei wurde darauf geachtet, dass möglichst wenig am bestehenden Code geändert wird. Eine weitere Voraussetzung war, das dieser auch ohne HAP4NfSen Plugin weiterhin funktioniert.
Bei der oben gezeigten Tabelle handelt es sich um die Ausgabe der Kommandozeilen Applikation NfDump, welche als Backend für NfSen dient und auch vom HAP4NfSen Plugin verwendet wird. NfSen erweitert standardmässig den Output um eine Whois Lookup Funktion, indem mit einer Regular Expression die IP(falls vorhanden) mit einem Stück JavaScript Code ergänzt wird.
Die vom HAP4NfSen erweiterte Version macht sich zu Nutze, dass die verwendete PHP Funktion preg_replace auch in einer überlandenen Version, welche einen Array von Pattern und Replacements verarbeitet, existiert. Der Code überprüft dabei, ob das HAP4NfSen Plugin geladen ist und erweitert in jenem Fall Patterns und Replacements. Dabei werden auch weitere Eigenschaften der NfSen-Umgebung beachtet, damit die unterschiedlichen Ausgabetypen(IPv4, IPv4:Port usw.) unterstützt werden können.

1.3.2 SVG - JavaScript

Das SVG Format, welches im Plugin zur Darstellung der Graphlets verwendet wird, kann mit Hilfe von JavaScript dynamisch angepasst werden. Zur manipulation der SVG Graphlets wird SVGPan, eine JavaScript Library, eingesetzt. Für den Einsatz im Hap4NfSen Plugin musste die Library erweitert werden.
Bei der SVGPan Library handelt es sich um eine .js Datei. Um die gewünschte Funktionalität einer SVG Datei hinzuzufügen, muss in der Datei ein zusätzlicher Tag vorhanden sein, welcher die JavaScript Library referenziert. Daneben sind noch einige Anpassungen an anderen von GraphViz generierten SVG Attributen nötig, damit die Library wie gewünscht funktioniert.
Bei grössen Graphen werden die JavaScript Erweiterungen aus Performancegründen deaktiviert.

Highlighting von Kanten

Das Highlighting von Kanten des Graphes wird mit einer Erweiterung der SVGPan Library erreicht. Ein Listener, welcher auf Maus Ereignisse auf Elementen der SVG Graphik reagiert, überprüft, ob es sich beim geklickten Element um

eine Linie handelt.
Falls dies der Fall ist, werden die aktuellen Attribute zwischengespeichert. Danach wird eine neue Farbe gesetzt und die Breite der Linie wird erhöht.
Wenn zuvor bereits eine andere Linie hervorgehoben war, so werden deren ursprünglichen Attribute wiederhergestellt.

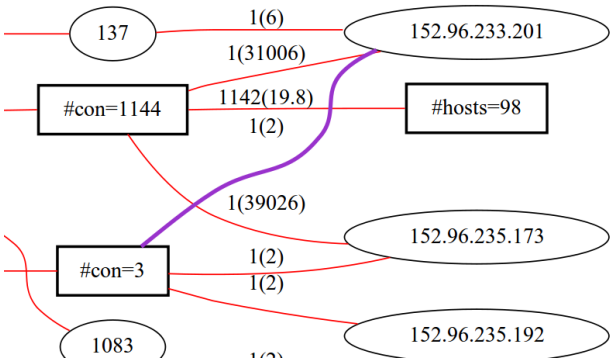

Figure 5: Eingefärbte Kante eines Graphlets

Zoom und Pan

Um das Graphlet um Zoom und Pan Funktionalität zu erweitern, wird die JavaScript Library SVGPan innerhalb der SVG Datei referenziert.
Dieses Library empfängt Maus Ereignisse (Klicks, Bewegungen sowie die Verwendung des Mausrades) und passt die SVG Datei dynamisch an. Da alle Informationen in der Vektorgraphik enthalten sind, muss dazu zu keiner Zeit eine Anfrage an den Server geschickt werden.

1.3.3 PHP - Perl

NfSen bietet allen Plugins eine Möglichkeit zum einfachen Datenaustausch zwischen dem PHP Frontend und dem Perl Backend an. Details dazu sind im NfSen Plugin Writers Guide[5] beschrieben.
Um Kollisionen mit bereits verwendeten Parameter Namen zu verhindern, wird jeweils für jeden der Parameter das Prefix "hap4nfsen_" verwendet.

Bereitstellung der Daten

Die PHP Funktion generateNetflowFile ist für den Aufruf der Backend Funktion assembleNetflowData, welche die NetFlow Daten zur Verfügung stellt, verantwortlich. Bei einem erfolgreichen Aufruf wird im HAP4NfSen Arbeitsverzeichnis eine neue Datei nach den spezifizierten Vorgaben erstellt und der Name dieser Datei wird an den Aufrufer zurückgegeben.

In Parameter

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_type	String	Alle von NfSen und NfDump unterstützten Profil Typen.	Der Type ist ein Parameter der den Typ des gewähltern Profils beschreibt. Der ausgewählte Wert stammt aus der NfSen Frontend Session und wird vom HAP4NfSen Plugin nicht verändert.
hap4nfsen_profile	String	Alle vorhandenen NfSen Profil Namen	Der Parameter stammt aus der NfSen Session und beschreibt, welches der vorhandenen Profile ausgewählt ist.
hap4nfsen_srcselector	String	Die angegebenen Datenquellen müssen in NfSen existieren. Falls mehrere Quellen angegeben sind, müssen diese durch ein Doppelpunkt (":") getrennt sein.	NfSen source, wie sie für den Aufruf von NfDump benötigt wird.
hap4nfsen_args	String	Es muss sich um erlaubte NfDump Aufrufargumente handeln.	Dieser Parameter enthält die Argumente, mit denen NfDump auf der Details Seite aufgerufen wurde.
hap4nfsen_filter	String	Der Wert muss ein gültiger NfDump Filter sein.	Enthält die Parameter, welche der Benutzer in der NfSen Details Seite in die Filter Box eingegeben hat.
hap4nfsen_hapfilter	String	Der Wert muss ein gültiger NfDump Filter sein.	Enthält zusätzliche Filter, die vom HAP4NfSen Plugin zur Darstellung des Graphlets generiert werden.
hap4nfsen_and_filter	String	Der Name eines in NfSen definierten Filters oder ein leerer String, falls kein Filter ausgewählt wurde	Der ausgewählte Wert enthält den Namen eines auf der NfSen Details-Seite angegebenen "and" Filters.

hap4nfsen_node_id_filters_X	String	Teil eines vollständigen NfDump Filters.	Dieser Parameter kann mehrmals vorkommen, wobei X dabei eine von 0 ausgehend durchgehend nummerierte Zahl ist. Der gesammte String enthält einen NfDump Filter, welcher dem Drilldown in einen Knoten entspricht.

Out Parameter

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_netflow_file	String	Name des generierten Files oder ein leerer String, falls der Aufruf erfolglos war.	Der Wert enthält den Namen der erstellten Datei im Arbeitsverzeichnis des Plugins. Dieser Dateiname kann als Parameter an weitere Backend Aufrufe weitergegeben werden.

Sequenz Diagram

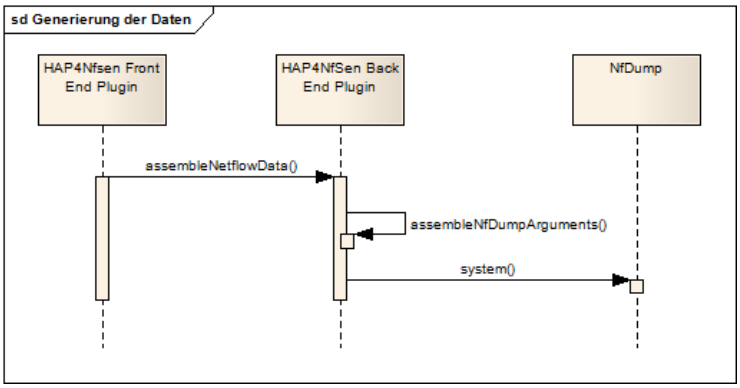

Figure 6: Vereinfachtes Sequenzdiagramm zur Bereitstellung der Daten

Nachdem die Funktion assembleNetflowData vom Frontend aufgerufen wurde, werden als erstes die übergebenen Parameter auf Vollständigkeit überprüft.
Danach wird die Funktion assembleNfDumpArguments aufgerufen, deren Aufgabe es ist, einen String zu erstellen, welcher alle nötigen Parameter für den Aufruf von NfDump enthält. Dazu gehört auch das Zusammenstellen des benötigten Filters, welcher danach in einer Datei im Arbeitsverzeichnis, welche als Parameter dem Aufruf mitgegeben wird, angegeben wird.
Der nächste Schritt ist der Aufruf von NfDump mit Hilfe des Perl system Befehls. Der Name der generierten Datei wird danach dem Aufrufer zurückgegeben.

Generierung der Graphendefinition

Für die Generierung der Graphendefinition ist im PHP Frontend die Funktion "generateDotFile" verantwortlich. Sie ruft die Perl Funktion "generateDotFile" des Backend Plugins auf.
Falls der Aufruf erfolgreich ist, wird eine neue Graphendefinition in Form einer .dot Datei im HAP4NfSen Arbeitsverzeichnis erstellt und der Name dieser Datei wird an den Aufrufer zurückgegeben. Die tempräre Datei mit den Input Rohdaten wird dabei gelöscht.

In Parameter

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_netflow_file	String	Eine im HAP4NfSen Arbeitsverzeichnis vorhandene Netflow-Datei	Dieser Parameter enthält die Rückgabe des NfDump-Aufrufs

hap4nfsen_ip	String	Eine IPv4 Addresse, die in Der NetFlow Datei, welche in hapviewer_netflow_file spezifiziert ist, vorhanden ist	Die in diesem Parameter angegebene IP Addresse wird im generierten Graphlet zur Host IP, welche zum einzigen Knoten in der ersten Partition(von links) des Graphlets wird.
hap4nfsen _summarization _client_roles	Boolean	True oder False.	Sagt aus, ob Client Rollen summarisiert werden sollen.
hap4nfsen _summarization _multi_client	Boolean	True oder False.	Sagt aus, ob Multiclient Rollen summarisiert werden sollen.
hap4nfsen _summarization _server_roles	Boolean	True oder False.	Sagt aus, ob Server Rollen summarisiert werden sollen.
hap4nfsen _summarization _p2p_roles	Boolean	True oder False.	Sagt aus, on Peer to Peer Rollen summarisiert werden sollen.
hap4nfsen_plugin_id	Integer	Zahlen von 0 bis N-1, wobei N = Anzahl der installierten Plugins .	Der Wert enthält die Plugin Id des Frontend Plugins, welche wiederum der Subtab-Id auf der NfSen Plugin Seite entspricht.

Out Parameter

Name	**Typ**	**Erlaubte Werte**	**Beschreibung**
hap4nfsen_dot_file	String	Dateiname der erstellten .dot Datei oder, im Falle eines Fehlers, ein leerer String	Der Namen der im Arbeitsverzeichnis erstellten .dot Datei wird hier zurückgegeben. Dieser Name ist nötig, um die SVG Graphik, welche dem Benutzer angezeigt wird zu generieren.
hap4nfsen_node _id_filters	Array	Elemente im Format "Node ID = NfDump Filter"	Zum Speichern in der PHP Session des Benutzers werden zu jeder generierten .dot Datei Filter zu allen Knoten zurück gegeben. Lange Filter können auf mehrere Einträge im Array aufgeteilt werden. Daher kann es für eine Node-Id mehrere Einträge geben.

hap4nfsen_node_id_summarization	Array	Elemente des Formats "Node ID = Rollenname "	Um die korrekten Rollensummarisierungen beim Drill Down aufzuheben, wird dem Frontend eine Liste mitgegeben, welche aussagt, an welchen Rollen ein Knoten beteiligt ist. Da ein Knoten an mehreren Summarisierungen beteiligt sein kann, kann es pro Node-Id auch mehrere Einträge im Array haben.
hap4nfsen_node_count	Integer	Die Anzahl von unterschiedlichen Node Ids der generierten Datei	Dieser Parameter wird vom Frontend verwendet um zu ermitteln, ob die Drilldown und Zoom Funktionalität ein- oder ausgeschaltet werden soll.

Sequenz Diagram

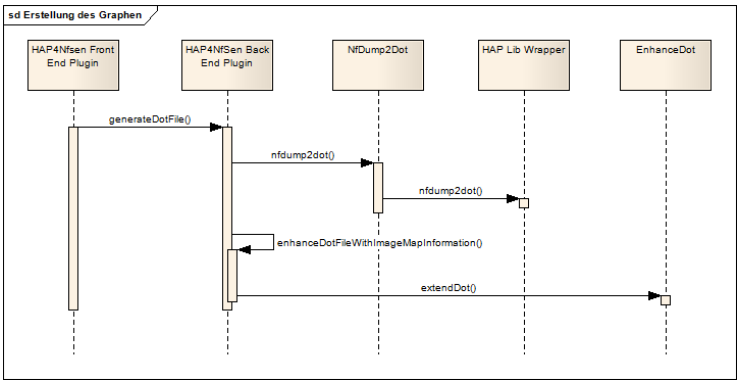

Figure 7: Vereinfachtes Sequenzdiagramm zur Generierung der Graphendefinition

Nachdem das Plugin im vorherigen Schritt eine temporäre NetFlow Datei im Arbeitsverzeichnis abgelegt wurde, ruft das Frontend die Backend Funktion generateDotFile auf.
Im nächsten Schritt wird das Modul NfDump2Dot aufgerufen, welches den Aufruf der HAP Library kapselt. Detailierte Informationen dazu sind unter den Perl - C/C++ Schnittstellen zu finden.
Nachdem die .dot Datei im Arbeitsverzeichnis erstellt wurde, wird die nicht mehr benötigte NetFlow Datei gelöscht. Danach wird die Funktion enhanceDot-FileWithImageMapInformation aufgerufen, welche ihrerseits das EnhanceDot Perl Modul aufruft.
In diesem wird die .dot Datei so erweitert, das das zukünftige Graphlet später klickbare Knoten enthält. Dazu werden Filterinformationen, welche vom der HAP Library in Form von Kommentaren in der Datei abgelegt wurden geparst und aufbereitet.

Erstellung des Graphlets

Für die Erstellung des Graphlets ist im Frontend Teil des Plugins die Funktion "generateGraphlet" verantwortlich, welche die gleichnamige Funktion des Backends aufruft.
Bei einem erfolgreichen Aufruf wird der zuvor von der HAP Library erstellte Graph in ein SVG Graphlet umgewandelt.

In Parameter

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_dot_file	String	Datei Namen einer .dot Datei	Dieser Wert bestimmt, welche Graphenbeschreibung mit Hilfe von GraphViz zu einer Graphikdatei umgewandelt wird.
hap4nfsen_disable_svg_js	Integer	1 oder 0	Dieser Parameter bestimmt, ob das Graphlet über Zoom Funktionalität verfügen soll. Bei 0 ist zoomen und verschieben möglich, bei 1 nicht.

Out Parameter

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_graphlet	String	Dateiname des generierten Graphlets oder ein leerer String	Dieser Rückgabewert enthält den Dateinamen der generierten Graphlet Datei im NfSen Bilder Verzeichnis. So kann das Bild aus dem Frontend via pic.php angezeigt werden. Falls die Generierung des Bildes feghlschlägt, wird ein leerer String zurück gegeben.

Sequenz Diagram

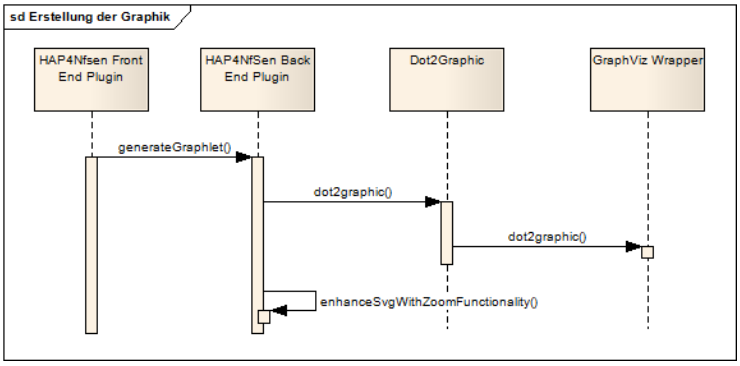

Figure 8: Vereinfachtes Sequenzdiagramm zur Erstellung des Graphlets

Die Funktion "generateGraphlet" wird vom Frontend aufgerufen. Danach erstellt das Backend mit Hilfe des Perl Moduls Dot2Graphic die SVG Datei. Details zur

Umwandlung von .dot zu .svg mittels GraphViz sind in den Schnittstellen unter "Perl - C/C++" zu finden.
Falls der Parameter zur Aktivierung der JavaScript Erweiterungen gesetzt ist, wird die erstellte SVG Graphik nochmals mit der Funktion "enhanceSvgWithZoomFunctionality" erweitert, bevor der Name der Datei dem Frontend Plugin zurückgegeben wird.

Lesen der Konfiguration

Um das Frontend Plugin mit den in der NfSen Konfigurationsdatei spezifizierten Parametern zu initialisieren, wird im PHP die Funktion "initPlugin" aufgerufen, welche die Funktion "getConfig" des Backends aufruft.
Mit Hilfe einer Variabel merkt sich der PHP Code, ob er die Konfiguration bereits gelesen hat. Dadurch wird die Backend Funktion nur ein Mal aufgerufen.
Zum Auslesen der Konfiguration steht im Backend eine Variabel der NfSen Umgebung zur Verfügung, mit der direkt auf Plugin Einstellungen zugegriffen werden kann.

In Parameter

Dieser Aufruf übergibt keine Parameter an das Perl Modul.

Out Parameter

Alle für das Frontend Plugin relevanten Informationen aus der Konfigurationsdatei werden bei der Rückgabe dieses Aufrufs mitgegeben.

Name	Typ	Erlaubte Werte	Beschreibung
hap4nfsen_max_history	String	Eine Zahl welche grösser oder gleich eins ist.	Der Parameter beschreibt die Anzahl der Schritte, die sich das Frontend zwecks Undo Funktion merken muss. Falls nicht definiert, wird ein Default Wert von 16 verwendet.

1.3.4 Perl - C/C++

Die verwendeten Tools GraphViz und HAP Viewer stehen in Form von Shared Libraries zur Verfügung. Um diese aus dem Perl Code des HAP4NfSen Plugin Backends aufrufen zu können, wurde SWIG eingesetzt, welches Perl Wrapper zum C/C++ Code erstellt.

GraphViz

GraphViz ist eine Visuallisierungssoftware, die vom HAP4NfSen Plugin zur Generierung von SVG Graphiken aus .dot Dateien verwendet wird. Die Library Version, welche als Shared Library zur Verfügung steht, wird vom Backend Plugin mit Hilfe von SWIG angesprochen. Die Perl Funktion, welche GraphViz aufruft heisst "dot2graphic".

In Parameter

Name	Typ	Erlaubte Werte	Beschreibung
type	String	Jedes von GraphViz unterstützte output Format(z.B. svg)	Verschiedene Ausgabeformate werden von der GraphViz Library unterstützt. In diesem Plugins wird aber nur SVG verwendet.
input	String	Pfad zu einer gültigen .dot Datei	Hier wird die .dot datei angegeben, welche zuvor von der HAP Library erstellt wurde. Diese Graphendefinition enthält alle Informationen zur Generierung des Graphlets.
output	String	Pfad, an der die generierte Datei hingeschrieben werden soll	Dieser Parameter gibt an, wohin das Graphlet geschrieben werden soll. Dies sollte an einem Ort sein, der für pic.php zugänglich ist. Ansonsten kann das Graphlet im Frontend nicht angezeigt werden.

Out Parameter

Name	Typ	Erlaubte Werte	Beschreibung
-	int	Eine Zahl zwischen -3 und +3.	Falls der Aufruf erfolgreich war, wird 0 zurück gegeben. Positive Zahlen stehen für fehlende input Parameter(z.B. 1 wird zurückgegeben, wenn der erste Parameter null ist). -1 Wird zurückgegeben, wenn die .dot Datei nicht gelesen werden kann. Der Wert -2 steht für Fehler bei der Generierung der Graphik und -3 für Probleme beim Schreiben der Ausgabedatei.

HAP Library

Die Erstellung von Graphlets wird von der Library Version des HAP Viewers[2], von dem es auch eine Fat Client Variante gibt, übernommen. Bei der Library handelt es sich um eine Shared Library, die in C++ geschrieben wurde und von uns aus Perl via SWIG aufgerufen wird.

In Parameter

Name	Typ	Erlaubte Werte	Beschreibung
input	String	Pfad und Name einer NfDump(Netflow) Datei	Enthält den vollständigen Namen einer zuvor mit Hilfe von NfDump generierten Netflow Datei im Arbeitsverzeichnis des Backend Plugins.
output	String	Pfad und Name der zu generierenden .dot Datei	Dieser Parameter gibt eine noch nicht existierende .dot Datei im Arbeitsverzeichnis des Backend Plugins an.
ip	String	Eine in den Netflows enthaltene IP(v4)	Beschreibt, welche IP zur Host Ip des Graphlets werden soll.
summarize_client_roles	Boolean	True oder False	Gibt an, ob Client Rollen summarisiert werden sollen.

summarize_multi _client_roles	Boolean	True oder False	Gibt an, ob Multi Client Rollen summarisiert werden sollen. Wird automatisch auf false gesetzt, falls Client Rollen nicht summarisizert werden.
summarize_server _roles	Boolean	True oder False	Gibt an, ob Server Rollen summarisiert werden sollen.
summarize_p2p_roles	Boolean	True oder False	Gibt an, ob Peer to Peer Rollen summarisiert werden sollen.

Out Parameter

Name	Typ	Erlaubte Werte	Beschreibung
-	Integer	Zahlen zwischen -1 und 3	Die Rückgabewerte 1 - 3 sind für nicht gesetzte Parameter(input, output, ip) reserviert. Beim erfolgreichen generieren einer .dot Datei wird 0 zurückgegeben. Der Wert -1 steht für einen Fehler, welcher in der HAP Library aufgetreten ist.

1.3.5 Perl - Unix

NfDump

NfDump wird wie zuvor beschrieben vom HAP4NfSen Backend Plugin eingesetzt, um die Input Daten für die Library Version des HAP Viewers zu erstellen. Der Aufruf der Kommandozeilen Applikation erfolgt mit dem Perl Kommando "system". Details zu den erlaubten Parametern sind auf der NfDump Website[1] zu finden.

1.4 Gewinnung der Summary-Nodes Information

Um eine Drilldown-Funktionalität anzubieten, muss das Plugin die Information, was für Rollen sich hinter jedem Knoten befinden, bereit halten. Da die HAP-Bibliothek in der "Vanilla"-Version diese Informationen weder in die temporäre, intern verwendete "hpg"-Datei noch in die schlussendlich produzierte .dot-Datei schreibt, musste die HAP-Bibliothek etwas erweitert werden. Ein vorhergehender Versuch, im Backend/Perl diese Informationen aus der .dot-Datei zurückzugewinnen erwies sich als sehr schwierig und unperformant.
Um diese Funktion nachzurüsten wurde die Singleton-Klasse "HAP4NFSENStore" eingeführt, in welche vor dem Schreiben der .hpg-Datei die benötigen, weitergehenden Informationen in eine Map hinterlegt werden. Als Schlüssel werden dazu die in die .hpg-Datei geschriebene Werte verwendet. Beim Schreiben der .dot-Datei können dann diese Informationen gefunden und ausgewertet werden. Beispiel eines am Ende der .dot-Datei eingefügten Kommentars:

```
/* Comments for HAP4NFSEN
 * k2_6=TCP:::
 * k2_17=UDP:::
 * k3_270008324=TCP:55115,55139,55148,55159:80:193.5.9.102:unibiflow_in:c
 * k3_268489074=TCP:53618::
 * k4_269156432=TCP:55264:80:74.125.232.112,193.5.9.102:biflow
 * k4_268828815=TCP:51227:143:
 * k5_4294836231=TCP:55124,55202:80:66.206.195.82,208.88.186.244:inflow:m
 * k5_4294836231=TCP:55199,55203:443:70.142.15.18,,208.88.186.244:inflow:m
 */
```

Jede Zeile ist wie folgt aufgebaut:

```
k<Partition>_<Nodenummer>=<Protokoll>:[<LocalPorts>]:
[<RemotePorts>]:[<RemoteIP>]:[<Flowdirection>][:<Summary-Type>]
```

Begriff	**Beschreibung**	**Mögliche Werte**
Partition:	Entspricht der Partitionsnummer	2 bis 5
Nodenummer:	Eindeutiger, nummerischer Wert des Knoten	unsigned Integerwert
Protokoll:	Enthält das Protokoll	TCP, UDP, ICMP, OTHER
LocalPorts:	Source-Ports, kommagetrennt falls mehrere	0-65535
RemotePorts:	Destination-Ports, kommagetrennt falls mehrere	0-65535
Flowdirection:	Gibt an, in welche Richtung der Flow zeigt	biflow, inflow, outflow, unibiflow_in, unibiflow_out
Summary-Type:	Gibt an, welche Art von Rollen (Server, Multiclient, Client, Peer2Peer) sich hinter dem Summary-Node befinden	s, m, c, p

1.4.1 Erweiterungen der HAP-Bibliothek

Verschlankung

Um von möglichst wenigen Bibliotheken abhängig zu sein, wurden gewisse Codestellen in bedingt kompilierte Code-Blöcke ausgelagert. Damit ist es möglich, auf die Installation von den Bibliotheken ipfix sowie pcap/pcap++ zu verzichten. Bei der Kompilierung der HAP-Library muss dazu dem Preprozessor die Variable "HAP4NFSEN_NFDUMPONLY" definiert werden. Eine weitergehende Reduzierung der Abhängigkeiten seitens der HAP-Library ist vorgesehen.

1.5 User Interface

In den folgenden Abschnitten werden alle Elemente der Hap4NfSen Plugin Seite beschrieben. Bereiche wie etwa der Titelbalken werden von NfSen selbst erstellt

und werden daher hier nicht weiter erklärt.

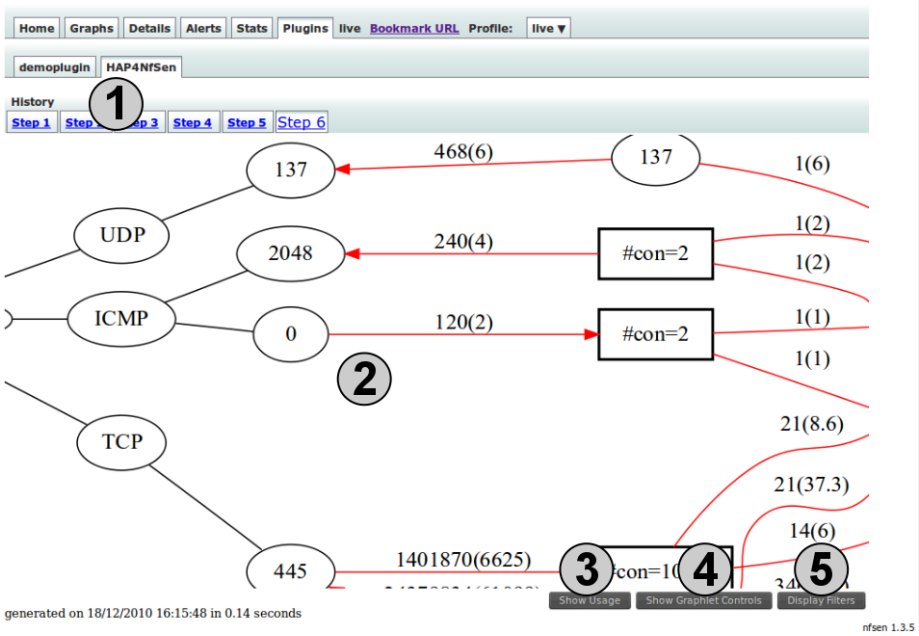

Figure 9: Elemente des Plugin User Interfaces

Folgende Elemente sind im Interface zu finden:

1. History Bar
2. Graphlet
3. Button "Show Usage"
4. Button "Show Graphlet Controls"
5. Button "Display Filters"

1.5.1 History Bar

Die History Bar ermöglicht es dem Benutzer vorhergänginge Aktionen rückgängig zu machen, indem er auf einen seiner letzten Schritte klickt.
Beim Einstieg auf die Plugin Seite steht nur ein einzelner Schritt zur Verfügung. Interaktionen mit dem Graphlet fügen dem Balken weitere Elemente zu.

Figure 10: Element "History Bar" des Benutzerinterfaces

Um diese Funktionalität zu implementieren verwendet das Plugin einen Stack, wobei in jedem Element alle Informationen gespeichert sind, welche benötigt werden, um das Graphlet zu generieren, sowie eine eindeutige ID.
Das oberste Element des Stacks(zugänglich durch Variable "Context", welches immer auf das oberste Element des Stacks zeigt) enthält dabei immer die aktuellen Daten.
Bei jedem Aufruf wird immer zuerst der History Stack angepasst und die Context Variable gesetzt. Bei gewöhnlichen Requests wird der Stack um ein Element erweitert. Falls es such um einen Undo Request handelt, wird als Parameter die eindeutige ID des Schrittes angegeben, zu welchem der Benutzer zurückspringen möchte.
Die einzige Manipulation, welche dazu nötig ist, ist das entfernen aller Elemente, welche über dem Element mit der angegebenen ID liegen.

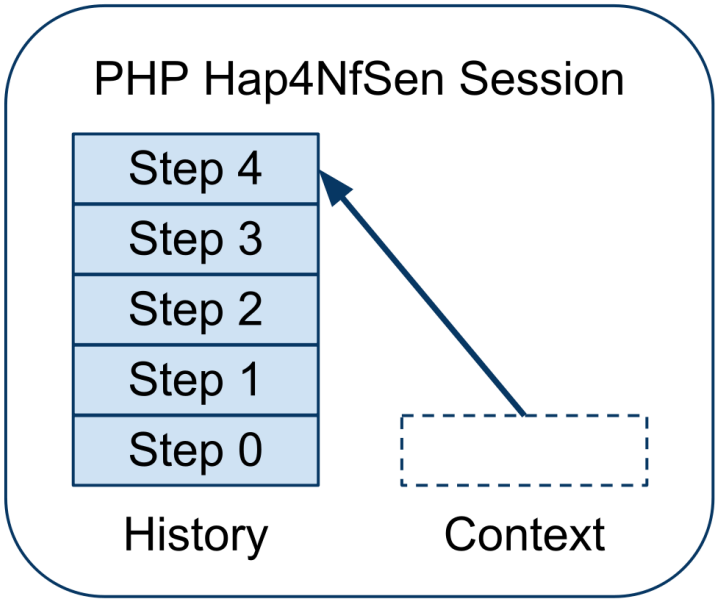

Figure 11: Implementation der Undo Funktionalität

1.5.2 Graphlet

Der grösste Teil der Plugin Seite wird vom HAP Graphlet, bei welchem es sich um eine SVG Graphik handelt, belegt. Die verschiedenen Möglichkeiten zur Interaktion mit den Graphlets sind im Abschnitt "Schnittstellen" ausführlich beschrieben.

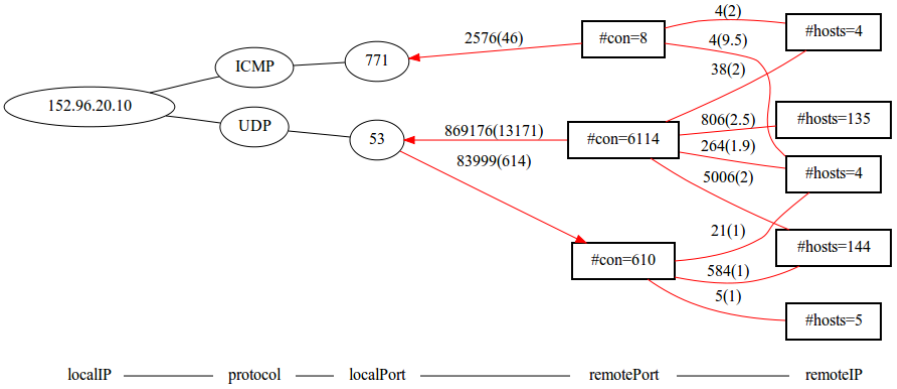

Figure 12: Beispiel eines Graphlets

1.5.3 Button "Show Usage"

Ein Klick auf diesen Button bringt die eingebaute Benutzeranleitung zum Vorschein und lässt den Button verschwinden. Weitere Informationen dazu sind im Abschnitt "Benutzeranleitung" zu finden.

1.5.4 Button "Show Graphlet Controls"

Per Klick auf den "Show Graphlet Controls" Button öffnet sich ein Bereich, mit dem die Generierung der Graphlets beeinflusst werden kann.
Um Probleme im Browser zu verhindern, werden standardmässig bei Graphen mit mehr als 128 Knoten die JavaScript Erweiterungen deaktiviert. In diesem Fall wird die vollstängige SVG Graphik angezeigt und Benutzer können durch scrollen des Browser Fensters navigieren. Bei noch grösseren Graphen(Anzahl Knoten grösser als 1024) wird die Graphik nicht mehr erstellt um eine Überlastung des Servers zu vermeiden.
Dieses Verhalten kann mit den beiden Buttons in diesem Bereich überschrieben werden.
Falls die JavaScript Funktionalität oder die Anzeige des Graphlet vom Plugin deaktiviert wurde, so wird der Bereich mit den Steuerelementen für das Graphlet automatisch angezeigt.

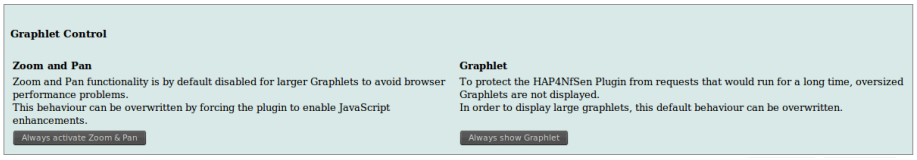

Figure 13: Bereich mit Elementen zur Kontrolle des Graphlets

1.5.5 Button "Display Filters"

Durch einen Klick auf diesen Button wird ein Bereich eingeblendet, in dem die aktiven Filter dargestellt werden.

Die angezeigten Filter und Summarisierungen dienen nur zur Information und können nicht direkt bearbeitet werden. Eine genaue Beschreibung der Bedeutung der Filter ist durch die Hilfefunktion des Hap4NfSen Plugins verfügbar.

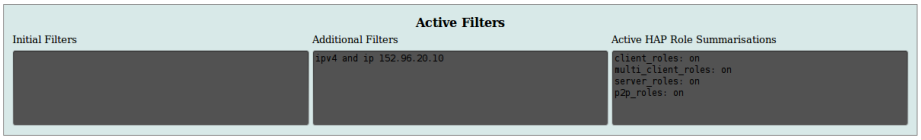

Figure 14: Aktuell verwendete Filter

1.6 Benutzeranleitung

Da die Bedienung des Plugins sehr einach ist, haben wir auf eine separate Benutzeranleitung verzichtet.
Eine kurze Beschreibung aller Funktionen wurde in das Frontend integriert, sodass diese den Benutzern jederzeit per Knopfdruck zur Verfügung steht.

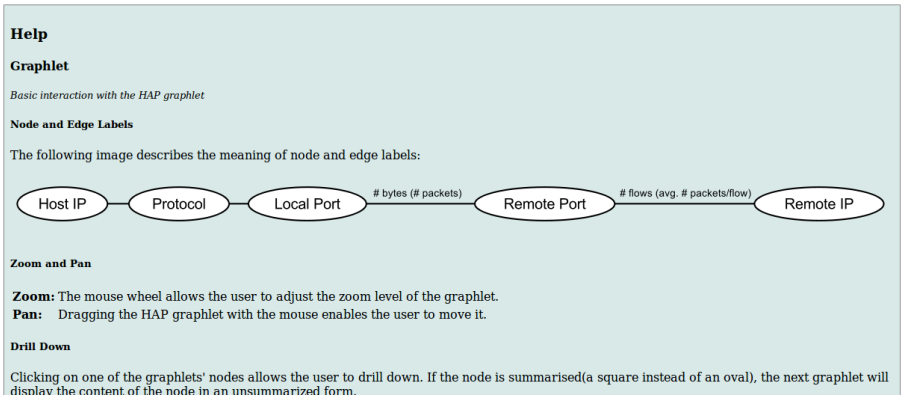

Figure 15: Hilfefunktion, wie sie im Hap4NfSen Frontend zu finden ist

1.7 Schlussfolgerungen

1.7.1 Zusammenfassung

In den ersten Wochen wurde ein Grossteil der Zeit in die Erstellung des Projektplans, sowie die Einarbeitung in die verschiedenen eingesetzen Technologien verwendet. Nach bereits kurzer Zeit hatten einen einfachen Prototyp erstellt, welcher grundlegende Aktionen, wie etwa das erstellen eines Graphlets aus Perl durchführen konnte. Danach erweiterten wir unseren Prototypen Schritt für Schritt um gewünschte Features.
Nach einer gewissen Zeit ergaben sich aber auch einige unvorhergesehene Probleme. Dazu gehörte zum Beispiel der Zugang zu Filterinformationen, welche hinter summarisierten Knoten stecken. Um an diese zu gelangen, musste die HAP Library erweitert werden. Auch gestaltete sich die Portierung des Plugins auf OpenBSD etwas umfang- und problemreicher als zuerst geplant.

Erfreulicherweise war es möglich, die Arbeit zu verschiedenen Zeitpunkten bei SWITCH zu präsentieren. Auf diese Weise konnten Probleme frühzeitig erkannt und Ideen für neue Features besprochen werden.

1.7.2 Resultat

Es gelang, alle wichtigen Features zu implementieren, sodass alle funktionalen Anforderungen erfüllt sind. Bei der Performance kann es bei sehr grossen Graphen zu Problemen mit der GraphViz Library kommen. In solchen Fällen wird das Graphlet nur auf expliziten Wunsch des Benutzers erstellt.
Da sich der genaue Umfang dieser Arbeit erst mit der Zeit und durch mehrere Besprechungen mit dem Betreuer sowie dem Industriepartner SWITCH ergeben hat, gestaltet sich ein Analyse des Resultates schwierig. Die positiven Äusserungen seitens von SWITCH/Peter Haag zeigen aber, dass das Resultat im erwarteten Rahmen liegt.

1.7.3 Mögliche Erweiterungen

Erweiterte Browser Unterstützung

Die aktuelle Version des Frontend Plugins unterstützt nur Mozilla Firefox als Web Browser vollständig. Wir haben uns für diesen Browser entschieden, da dieser für alle bekannten Platformen(Windows, Linux, Mac OS) verfügbar ist.
Für alle Benutzer, welche normalerweise einen anderen Browser einsetzen, wäre es sicher angenehmer, wenn sie diesen auch für das HAP Plugin verwenden könnten.
Daher sollten zukünftige Versionen des Plugins alle bekannten Browser(z.B. Opera und Apple Safari) unterstützen. Da Internet Explorer momentan noch keine SVG Graphiken, welche für das Graphlet benötigt werden, anzeigen kann, ist eine Unterstützung nicht oder nur mittels einem Plugin möglich.
Momentan sind Probleme mit folgenden Browsern bekannt:

Browser Name	**Problembeschreibung**
Opera für Linux	Zoom und Pan Funktionalität für Graphlets ist nicht verfügbar. Zudem dauert das rendern von grossen SVG Graphiken deutlich länger als in den anderen getesteten Browsern.
Safari für Windows, getestet unter Linux	Transparente Bereiche von SVGs werden weiss dargestellt. Dies führt zu einigen unschönheiten(z.B. in der Benutzeranleitung). Die Funktionalität des Plugins wird dadurch allerdings nicht eingeschränkt.

Filterung von ICMP Type und Code Filtern

Momentan werden ICMP Types und Code Filter, welche in NfSen und in den HAP Graphlets wie Ports der Protokolle TCP und UDP dargestellt werden, ignoriert, wenn sie als Filter verwendet werden.
In einer weiteren Version des Plugins könnte das Plugin so angepasst werden, das es die Type und Code Informationen in korrekte Filter umwandeln und diese anwenden kann.

Erweitertes Highlighting von Pfaden

Momentan kann man per Mauscklick jeweils eine Linie des HAP Graphlets Highlighten. Eine mögliche Erweiterung dazu wäre das Highlighting eines ganzen Pfades(die Linie zieht sich durch alle Partitionen des Graphen).
Diese Funktion würde es dem Benutzer vereinfachen, sich in grossen Graphen zurechtzufinden.

Bubble Up

Das Graphlet erlaubt es dem Benutzer im Moment per Click auf einen Knoten einen Drill Down auszuführen. Allerdings gibt es neben der History keine Möglichkeit, einen Drill Down wieder rückgängig zu machen.
Eine solche Funktion würde den Benutzern die Navigation durch das Graphlet weiter vereinfachen.

Filter beim Wechseln zur NfSen Details Seite behalten

Aktuell gelten die auf der Plugin Seite zusammengestellten Filter nur für das HAP Plugin. Eine mögliche Erweiterung wäre ein zusätzlicher Button, mit dem man zur NfSen Details Seite zurückwechseln und dabei die aktuellen Filter beibehalten kann.

Integration in NfSen 2.0

Eine vollständig überarbeitete Version von NfSen ist bereits in der Entwicklung. Sobald diese verfügbar ist, kann das Plugin an die neue Schnittstelle angepasst werden.
Voraussichtlich wird die neue Version AJAX einsetzen, was dem Hap4NfSen Plugin weitere Möglichkeiten bieten würde.

Deaktivierung einzelner Summarisierungen

Momentan lassen sich die vier Summarisierungsmöglichkeiten des HAPviewers nur für das komplette Graphlet ein- oder ausschalten. Das ist teilweise unerwünscht, da eine komplette Desummarisierung zu einem Verlust des Überblicks führt.
In einer späteren Version des HAPviewers könnte daher eine weitere Funktion hinzugefügt werden, welche es erlaubt, ausgewählte Knoten zu desummarisieren.

Rückgabewert des NfDump Aufrufs

Momentan zeigt der Rückgabewert des NfDump Aufrufs im Perl Backend einen Fehler an. Dieser tritt auf, obwohl die Erstellung der NetFlow Datei funktioniert.

Der Code des Plugins ignoriert den Rückgabewert nun, was unschön ist. In einer späteren Version des Plugins sollte dies untersucht und korrigiert werden.

Darstellungsfehler mit der Kombination OpenBSD und Firefox/Chrome

Aus einem uns noch unbekannten Grund führt die Kombination des SWITCH-Testimages, welches um unser Plugin ergänzt wurde, sowie Firefox oder Chrome zu einem Darstellungsfehler. Den Grund dafür kennen wir noch nicht, auf mehreren anderen Testinstallationen oder aber bei Verwendung von Opera konnte der Effekt nicht beobachtet werden.

1.7.4 Weiteres Vorgehen

Im folgenden Semester wird es eine weiterführende Arbeit im selben Themengebiet geben. Eventuell können dabei einige der zuvor erwähnten Erweiterungen umgesetzt werden.

Part II
Berichtsanhang

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

INFORMATIK

Aufgabenstellung zur Studienarbeit HS 2010

„Integration des HAPviewers in das NfSen Framework“

Gruppe: Sebastian Hügli / Reto Schneider

Ausgangssituation

HAPviewer (host application profile viewer) ist ein Open-Source Tool, das an der ETH Zürich für die verhaltensbasierte Analyse des Netzwerkverkehrs einzelner Rechner entwickelt wurde. Es benutzt eine graphenbasierte Darstellung, die sich an das Berkeley Socketmodell anlehnt, sowie Summarisierungs- und Filtertechniken, die eine kompakte Darstellung von hunderten oder gar tausenden von Netzwerkverbindungen ermöglicht.
NfSen ist ein Open-Source Web-Frontend zu den *NfDump Tools* und erlaubt es einem Netzwerkspezialisten über unterschiedliche Diagramme den Verkehr ganzer Netze über eine Web-Oberfläche zu überwachen (http://nfsen.sf.net).
Die Integration des *HAPviewers* in *NfSen* verspricht eine attraktive Erweiterung der Einsatzmöglichkeiten dieser zwei Tools.

Aufgabe

Im Rahmen dieser Arbeit soll eine Bibliotheksversion des Stand-Alone Tools HAPviewer in das NfSen Framework integriert werden. Durch diese Erweiterung, die als *NfSen-Plugin* zu realisieren ist, wird es möglich das *Host Application Profile (HAP) Graphlet* einer durch den Benutzer ausgewählten IP-Adresse zu visualisieren. Ferner soll das Plug-In diese Graphlets mit der Maus klickbar machen, so dass zu jedem Knoten des Graphen zusätzliche Information (z.B. Diagramme) angezeigt werden können.

Im Einzelnen bedingt dies das Kennenlernen des NfSen Frameworks und seiner Plugin-Schnittstelle (http://nfsen.sourceforge.net/PluginGuide/plugin-guide.html), und die Realisierung mehrerer Perl-Module und PHP-Scripts, die u.a. die Bibliotheksversion des HAPviewers über eine C++ Schnittstelle anbinden.

Berichtsgestaltung

Der Bericht ist gemäss [1] zu gestalten.

[1] Eduard Glatz, „Vorgaben zur Berichtserstellung“, Ausgabe des 23. September 2010

Termine

20. Sept. 2010	Arbeitsbeginn
23. Dez. 2010	Abgabe des Berichts an den Betreuer (bis 17:00)

Weitere Termine siehe Terminangaben auf dem HSR-Web (intern).

Betreuung

Betreuer: Prof. Eduard Glatz, Email: eglatz@hsr.ch
Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt.

Industriepartner:

SWITCH, Werdstrasse 2, Postfach, 8021 Zürich, web: www.switch.ch

Kontaktperson:

Peter Haag, Email: peter.haag@switch.ch

Rapperswil, den 23. Sept. 2010

Eduard Glatz

2 Projektplan

2.1 Meilensteine

Nummer	**Name**	**Woche**	**Beschreibung**
1	Kick-Off-Meeting	1, Beginn	Besprechung von Details mit P. Haag.
2	Projektplan erstellt	2, Ende	Projekt- und Zeitplan erstellt
3	Anforderungsspezifikation erstellt	3, Ende	Ende Analyse
4	Prototypen erstellt	6, Ende	Ende Elaboration, Sitzung mit Switch, Ideen zeigen
5	1. Installation auf SWITCH Server	11	Erster Live-Betrieb
6	2. Installation auf SWITCH Server	12	Zweiter Live-Betrieb, gröbste Bugs behoben
7	Abgabe	14, Ende	Finish der Dokumentation, ePaper, Plakat, etc

2.2 Interationen

Iteration	**Inhalt**	**Start**	**Ende**
Inception	Projekt-/Zeitplanung, Startmeeting	Woche 1	Woche 2
Elaboration 1	Use-Cases, Anforderungsspezifikation	Woche 3	Woche 4
Elaboration 2	Design, Prototypen der Einzelnen Probleme	Woche 5	Woche 6
Construction 1	Erste Sitzung bei Switch, Umsetzung der gewählten Varianten	Woche 7	Woche 8
Construction 2	Ausbau der gewählten Varianten	Woche 9	Woche 10
Construction 3	Zweite und dritte Sitzung bei Switch, vorstellen und testen des aktuellen Stands, Feedback sammeln, weiterer Ausbau der gewählten Varianten	Woche 11	Woche 12
Transistion	Abschliessen/erstellen der Doku (Arbeitsdoku, ePaper, pers. Bericht, etc.), präparieren zum Veröffentlichen, Abgabe	Woche 13	Woche 14

2.3 Zeitplan

Siehe folgende Seite.

Zeitplan

Hap4NfSen

Phasen			Inception								Elaboration 1							
Meilensteine			MS1								MS2				MS3			
Woche (von - bis)			WOCHE 1 20.09		24.9.		WOCHE 2 27.09		10.01		WOCHE 3 04.10		08.10		WOCHE 4 11.10		15.10	
	Total		Seb		Reto		Seb		Reto		Seb		Reto		Seb		Reto	
Task	SOLL	IST	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I
Projekt Managment	48	37.2	2.0	2.0	2.0	2.0	9.0	8.0	9.0	7.0	2.5	2.5	2.5	2.5	1.5	1.0	1.5	0.5
Projektplan	18	14	1.0	1.0	1.0	1.0	3.0	3.0	3.0	2.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	
Zeitplan	25	22	1.0	1.0	1.0	1.0	5.0	5.0	5.0	5.0	1.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5
Risikoabschätzung	5	1.2					1.0		1.0		0.5	0.5	0.5	0.5				
Requirements	21.5	8.9									3.5	3.5	3.5	0.5	1.5	1.5	3.0	1.4
Use Cases essential	9.5	2.4									1.5	1.5	1.5	0.2	0.5	0.5	1.0	0.2
Use Case fully dressed	7.5	3.3									0.5	0.5	0.5	0.1	0.5	0.5	1.0	0.2
Anforderungsspezifikation	4.5	3.2									1.5	1.5	1.5	0.2	0.5	0.5	1.0	1.0
Analyse	29	13.75									1.5	1.5	1.5	0.5	3.0	2.0	3.0	0.3
Domainmodell	4	2									1.0	1.0	1.0		0.5	1.0	0.5	
Sequenzdiagramm	14	5.7													1.5	0.5	1.5	0.2
Operation Contracts	12	5.55													1.0	0.5	1.0	0.1
Use Case Diagramm	3	1									0.5	0.5	0.5	0.5				
Design	14	2.55																
Design Modell	10	1.5																
Logische Architektur	4	1.05																
Implementation	199.5	285									3.0	3.5	4.0	9.0	9.0	10.5	6.0	12.5
Prototyp Perl ↔ hapviz	8.5	12											1.0	2.5	1.0	1.0	3.0	4.0
Prototyp C++ ↔ Perl	8	10.5									1.0	1.0	1.0	5.0	1.0		3.0	3.0
Prototyp Perl ↔ PHP	7.5	10.5													2.0	2.0		
Prototyp Einstiegstpunkt	4	5.5										1.5						
Prototyp Bildanzeige	2	4														1.0		
Prototyp clickable-Bild	4	4									1.0	0.5	1.0	0.5				
manipulieren	10	10																5.0
Prototyp Graphlet in NfSen	5	6.5													5.0	6.0		0.5
Prototyp perl ↔ GraphViz	3.5	3.5									1.0	0.5	1.0	1.0				
Implementation	129	189.5																
Refactoring	18	29														0.5		
Test	24	11.5																
Usability Tests	12	4																
Browser Kompatibilitätstest	1.5	3																
Nichtfunktionale Anforderungen(Geschwindigkeit)	10.5	4.5																
Dokumentation	97.5	115.95	8.0	9.0	3.0	3.5	3.0	7.0		3.0	4.0	4.0	3.0	0.6	1.0	1.0	1.0	1.0
Man-Pages(falls nötig)	1																	
Anleitung	18	47.7	1.0	1.0	1.0	1.0		1.0		1.0	1.0	1.0	1.0	0.2				
Wiki-Seite	21	12.65	1.0	1.0	1.0	1.0		1.0		1.0	1.0	1.0	1.0	0.3	0.5	0.5	0.5	0.5
LaTeX	40	44.6	6.0	7.0	1.0	1.5	3.0	5.0		1.0	2.0	2.0	1.0	0.1	0.5	0.5	0.5	0.5
ePaper	8	5																
Plakat	10	6																
Studium Technologien	53.5	53.8	4.0	4.0	8.0	8.0	6.0	5.0	9.0	8.5	5.0	5.0	5.0	1.8	4.5	6.0	2.0	6.0
Perl	9	8.5									1.0	1.0	1.0	0.5	1.0	1.0	1.0	2.0
PHP	5	4.5													1.0	1.0		
NfSen/NfDump	15	12	4.0	4.0	3.0	3.0	3.0	2.0	3.0	1.0	1.0	1.0	1.0	0.5		0.5		
C/C++ to Perl Schnittstelle	12	15.75			5.0	5.0			3.0	2.5	2.0	2.0	2.0	0.5		2.0		3.8
(Auto)-Builds	4	5.3							3.0	5.0			1.0	0.3				
LaTeX	4	4					3.0	3.0			1.0	1.0						
(klickbare)-Bilder	1	0.5													1.0	0.5		
GraphViz	2.5	2.75													0.5	0.5	1.0	0.3
JavaScript	1	0.5													1.0	0.5		
Sitzungen	44.8	40.8	2.2	2.5	2.2	2.5	1.2		1.2		1.2	1.7	1.2	1.7	2.2	1.7	2.0	1.5
Team-Meetings	40	37.6	2.0	2.3	2.0	2.3	1.0		1.0		1.0	1.5	1.0	1.5	2.0	1.5	2.0	1.5
Protokoll	4.8	3.2	0.2	0.2	0.2	0.2	0.2		0.2		0.2	0.2	0.2	0.2	0.2	0.2		
Qualitätssicherung	1	1							1.0	1.0								
git	1	1							1.0	1.0								
Wochen Subtotal	532.8	570.5	16.2	17.5	15.2	16.0	19.2	20.0	20.2	19.5	20.7	21.7	20.7	16.6	22.7	23.7	18.5	23.2
Wochentotal	532.8	570.5			31.4	33.5			39.4	39.5			41.4	38.3			41.2	46.9

Phasen	Elaboration 2								Construction 1							
Meilensteine									MS4							
Woche (von - bis)	WOCHE 5 18.10		22.10		WOCHE 6 25.10		29.10		WOCHE 7 01.11		05.11		WOCHE 8 08.11		12.11	
	Seb		Reto		Seb		Reto		Seb		Reto		Seb		Reto	
Task	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I
Projekt Managment	1.5	1.0	1.0	0.6					1.5	0.5	1.0	1.6				
Projektplan	0.5	1.0	0.5	0.5					0.5		0.5	1.0				
Zeitplan	0.5		0.5						0.5	0.5	0.5	0.5				
Risikoabschätzung	0.5			0.1					0.5			0.1				
Requirements	1.0		1.0						1.0	0.5	1.0					
Use Cases essential	0.5		0.5						0.5		0.5					
Use Case fully dressed	0.5		0.5						0.5	0.5	0.5					
Anforderungsspezifikation																
Analyse	3.0	1.0	3.0	0.5					2.0	1.5	2.0	1.5				
Domainmodell	0.5		0.5													
Sequenzdiagramm	1.0	0.5	1.0	0.5					1.0	1.0	1.0	0.5				
Operation Contracts	1.0	0.5	1.0						1.0	0.5	1.0	1.0				
Use Case Diagramm	0.5		0.5													
Design	4.0	0.6			2.0	1.0	2.0	1.0					1.0		1.0	
Design Modell	2.0	0.5			1.0	0.5	1.0	0.5					1.0		1.0	
Logische Architektur	2.0	0.1			1.0	0.5	1.0	0.5								
Implementation	4.5	10.0	13.5	8.5	12.0	12.0	12.5	14.0	11.0	12.5	11.0	3.5	14.0	16.5	14.0	7.0
Prototyp Perl ↔ hapviz			1.5	2.0			2.0	2.5								
Prototyp C++ ↔ Perl							2.0	1.5								
Prototyp Perl ↔ PHP	0.5	4.0	1.0	0.5	2.0	2.0	2.0	2.0								
Prototyp Einstiegstpunkt	2.0	2.0			2.0	2.0										
Prototyp Bildanzeige	1.0	2.5			1.0	0.5										
Prototyp clickable-Bild	1.0	1.5			1.0	1.5										
manipulieren			10.0	5.0												
Prototyp Graphlet in NfSen																
Prototyp perl ↔ GraphViz			1.0	1.0			0.5	1.0								
Implementation					6.0	6.0	6.0	7.0	10.0	11.5	10.0	3.0	13.0	15.5	13.0	5.0
Refactoring									1.0	1.0	1.0	0.5	1.0	1.0	1.0	2.0
Test	0.5	0.5	0.5	0.5									3.0	1.5	3.0	2.5
Usability Tests													2.0	1.0	2.0	1.5
Browser Kompatibilitätstest	0.5	0.5														
Nichtfunktionale Anforderungen(Geschwindigkeit)			0.5	0.5									1.0	0.5	1.0	1.0
Dokumentation	1.0	0.7	1.0	2.1	1.0	0.5	1.0	0.5	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
Man-Pages(falls nötig)																
Anleitung																
Wiki-Seite	0.5	0.2	0.5	0.1	0.5		0.5	0.5	0.5	0.5	0.5	1.0	0.5		0.5	0.5
LaTeX	0.5	0.5	0.5	2.0	0.5	0.5	0.5		0.5	0.5	0.5		0.5	0.5	0.5	0.5
ePaper																
Plakat																
Studium Technologien	1.0	2.0	2.0	2.0	1.0	0.5	1.5	1.0	1.0	1.5	1.5	1.0	1.0	1.0	1.0	0.5
Perl		1.0	1.0			0.5	1.5	1.0			1.5	1.0			1.0	0.5
PHP	1.0	1.0			1.0				1.0	1.5			1.0	1.0		
NfSen/NfDump																
C/C++ to Perl Schnittstelle																
(Auto)-Builds																
LaTeX																
(klickbare)-Bilder																
GraphViz			1.0	2.0												
JavaScript																
Sitzungen	2.2	2.2	2.0	2.0	2.2	2.2	2.0	2.0	3.2	2.2	3.0	2.0	1.2	1.2	1.2	1.2
Team-Meetings	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	3.0	2.0	3.0	2.0	1.0	1.0	1.0	1.0
Protokoll	0.2	0.2			0.2	0.2			0.2	0.2			0.2	0.2	0.2	0.2
Qualitätssicherung																
git																
Wochen Subtotal	18.7	18.0	24.0	16.2	18.2	16.2	19.0	18.5	20.7	19.7	20.5	10.6	21.2	20.7	21.2	12.2
Wochentotal			42.7	34.2			37.2	34.7			41.2	30.3			42.4	32.9

Phasen	Construction 2								Construction 3							
Meilensteine									MS5				MS6			
Woche (von - bis)	WOCHE 9 15.11		19.11		WOCHE 10 22.11		26.11		WOCHE 11 29.11		03.12		WOCHE 12 06.12		10.12	
	Seb		Reto		Seb		Reto		Seb		Reto		Seb		Reto	
Task	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I
Projekt Managment	2.0	1.5	2.0	1.5					1.0		1.0	1.5				
Projektplan	0.5	0.5	0.5	0.5					0.5		0.5	0.5				
Zeitplan	1.0	1.0	1.0	1.0					0.5		0.5	1.0				
Risikoabschätzung	0.5		0.5													
Requirements	2.0	0.5	2.0	0.5					1.0	0.5	1.0					
Use Cases essential	1.0		1.0						0.5		0.5					
Use Case fully dressed	1.0	0.5	1.0	0.5					0.5	0.5	0.5					
Anforderungsspezifikation																
Analyse	2.5	1.5	2.5	1.5					2.5	1.0	2.5	1.0				
Domainmodell																
Sequenzdiagramm	1.0	0.5	1.0	0.5					1.0	0.5	1.0	0.5				
Operation Contracts	1.0	1.0	1.0	0.5					1.0	0.5	1.0	0.5				
Use Case Diagramm	0.5		0.5	0.5					0.5		0.5					
Design					1.0		1.0						1.0		1.0	
Design Modell					1.0		1.0						1.0		1.0	
Logische Architektur																
Implementation	9.0	12.0	9.0	33.5	14.0	14.5	9.0	25.0	9.0	11.5	9.0	11.0	13.0	14.5	13.0	16.0
Prototyp Perl ↔ hapviz																
Prototyp C++ ↔ Perl																
Prototyp Perl ↔ PHP																
Prototyp Einstiegstpunkt																
Prototyp Bildanzeige																
Prototyp clickable-Bild																
manipulieren																
Prototyp Graphlet in NfSen																
Prototyp perl ↔ GraphViz																
Implementation	8.0	10.5	8.0	33.0	13.0	14.0	8.0	24.0	8.0	10.5	8.0	10.0	9.0	13.0	9.0	14.0
Refactoring	1.0	1.5	1.0	0.5	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	4.0	1.5	4.0	2.0
Test					1.0		1.0		3.5	1.5	3.0	1.5	4.5	1.0	4.0	1.5
Usability Tests									2.0	0.5	2.0	0.5	2.0		2.0	0.5
Browser Kompatibilitätstest									0.5	1.0			0.5	0.5		
Nichtfunktionale Anforderungen(Geschwindigkeit)					1.0		1.0		1.0		1.0	1.0	2.0	0.5	2.0	1.0
Dokumentation	1.0	0.5	1.0	1.0	1.0	0.5	6.0	18.6	1.0	1.0	1.0	21.0	1.0	1.5	1.0	0.5
Man-Pages(falls nötig)																
Anleitung							5.0	18.0				20.0				
Wiki-Seite	0.5		0.5	0.5	0.5		0.5	0.1	0.5	0.5	0.5	0.5	0.5		0.5	0.5
LaTeX	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1.5	0.5	
ePaper																
Plakat																
Studium Technologien																
Perl																
PHP																
NfSen/NfDump																
C/C++ to Perl Schnittstelle																
(Auto)-Builds																
LaTeX																
(klickbare)-Bilder																
GraphViz																
JavaScript																
Sitzungen	1.2	1.2	1.2	1.0	1.2	1.2	1.2	1.0	1.2	1.2	1.2	1.0	1.2	1.2	1.2	1.0
Team-Meetings	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Protokoll	0.2	0.2	0.2		0.2	0.2	0.2		0.2	0.2	0.2		0.2	0.2	0.2	
Qualitätssicherung																
git																
Wochen Subtotal	17.7	17.2	17.7	39.0	18.2	16.2	18.2	44.6	19.2	16.7	18.7	37.0	20.7	18.2	20.2	19.0
Wochentotal			35.4	56.2			36.4	60.8			37.9	53.7			40.9	37.2

Phasen	Transisiton							
Meilensteine					MS7			
Woche (von - bis)	WOCHE 13 13.12		17.12		WOCHE 14 20.12		23.12	
	Seb		Reto		Seb		Reto	
Task	S	I	S	I	S	I	S	I
Projekt Managment	2.0	0.5	3.0	1.0	1.0	1.0	1.0	1.0
Projektplan	1.0		1.0	0.5				
Zeitplan	1.0	0.5	2.0	0.5	1.0	1.0	1.0	1.0
Risikoabschätzung								
Requirements								
Use Cases essential								
Use Case fully dressed								
Anforderungsspezifikation								
Analyse								
Domainmodell								
Sequenzdiagramm								
Operation Contracts								
Use Case Diagramm								
Design								
Design Modell								
Logische Architektur								
Implementation		7.0		14.0		1.5		5.0
Prototyp Perl ↔ hapviz								
Prototyp C++ ↔ Perl								
Prototyp Perl ↔ PHP								
Prototyp Einstiegstpunkt								
Prototyp Bildanzeige								
Prototyp clickable-Bild								
manipulieren								
Prototyp Graphlet in NfSen								
Prototyp perl ↔ GraphViz								
Implementation		3.5		5.0				4.0
Refactoring		3.5		9.0		1.5		1.0
Test		0.5		0.5				
Usability Tests								
Browser Kompatibilitätstest		0.5		0.5				
Nichtfunktionale Anforderungen(Geschwindigkeit)								
Dokumentation	12.0	8.5	12.0	3.0	14.5	13.5	15.0	10.0
Man-Pages(falls nötig)			0.5				0.5	
Anleitung	1.5	0.5	1.5	2.0	3.0	0.5	3.0	1.5
Wiki-Seite	1.5		1.5		2.5	0.5	2.5	1.0
LaTeX	4.5	5.0	4.5	1.0	4.5	8.0	4.5	4.0
ePaper	2.0	1.0	2.0		2.0	2.5	2.0	1.5
Plakat	2.5	2.0	2.5		2.5	2.0	2.5	2.0
Studium Technologien								
Perl								
PHP								
NfSen/NfDump								
C/C++ to Perl Schnittstelle								
(Auto)-Builds								
LaTeX								
(klickbare)-Bilder								
GraphViz								
JavaScript								
Sitzungen	1.2	1.2	1.2	1.0	1.2	1.7	1.2	1.5
Team-Meetings	1.0	1.0	1.0	1.0	1.0	1.5	1.0	1.5
Protokoll	0.2	0.2	0.2		0.2	0.2	0.2	
Qualitätssicherung								
git								
Wochen Subtotal	15.2	17.7	16.2	19.5	16.7	17.7	17.2	17.5
Wochentotal			31.4	37.2			33.9	35.2

2.4 Projektrisikoabschätzung

Nachfolgend werden Risiken aufgelistet und bewertet. Am Ende der 2. Iteration der Elaboration wurden die Risikopunkte im Team besprochen. Dabei wurden Risiken, die nicht mehr auftreten, können abgehakt und die verbleibenden Risiken wurden neu beurteilt.

Nr.	**Risiko**	**Vorbeugende Massnahmen und Workaround**	max. in h	p	p*h
R01	C/C++-Funktionsaurufe aus Perl funktionieren nicht wie gewünscht	Rechtzeitig Machbarkeit abklären, allenfalls Standalone in Betracht ziehen	10	0.1	1
R02	Datenmenge ist zu viel, um in akzeptabler Geschwindigkeit bearbeitet zu werden.	Früh perf. Tests durchführen, allenfalls eine Möglichkeit, um die Dateien vorgängig zu filtern finden	30	0.2	6
R03	Funktionen von NfSen können mittels HAPlib abgebildet werden	Früh Tests durchführen, Erkenntnisse an Herr Glatz mitteilen	10	0.5	5

2.5 Arbeitsaufteilung

Der Umfang des Projekts, sowie die Tatsache, das mehrere von einander getrennte Komponenten entwickelt wurden, bot an, die benötigten Arbeiten aufzuteilen. In den folgenden Abschnitten steht beschrieben, wer für welche Teile der Applikation verantwortlich war. Alle nicht ausdrücklich erwähnten Arbeiten wurden von beiden Studenten bearbeitet.

2.5.1 Reto Schneider

Bei der grössten Aufgabe, um welche sich Reto Schneider kümmerte handelte es sich um die Erweiterung der HAP Library inklusive der Ausgabe von NodeId Informationen als Kommentare in die .dot Dateien. Zuvor arbeitete er an einem anderen Ansatz, bei welchem versucht wurde, Filterinformationen direkt aus den .dot Dateien zu lesen. Dieser Code wird nicht mehr benötigt und ist daher in der finalen Version des Plugins nicht mehr enthalten.
Daneben erstellte er die SWIG Module, welche es erlauben den C/C++ Code der GraphViz und HAP Libraries aus Perl aufzurufen. Ausserdem führte er die portierung des Plugins auf OpenBSD durch.
Der Umfang all dieser Arbeiten beträgt in etwa 150 Arbeitstunden.

2.5.2 Sebastian Hügli

Sebastian Hügli erstellte den Grossteil des NfSen Front- und Backend Plugins. Dazu gehörte die Erstellung des PHP Web Interfaces, die Kommunikation mit

dem Backend, das Aufrufen der gekapselten C/C++ Funktionen, sowie der Umgang mit verschiedenen Filtern.
Zusätzlich erweiterte er die JavaScript Library SVGPan und erstellte den Einstiegspunkt auf der NfSen "details" Seite.
Insgesammt betrug der Aufwand für diese Arbeiten ebenfalls etwa 150 Arbeitsstunden.

3 Build Anleitung

3.1 Annahme

Diese Anleitung geht von einem System aus, auf dem NfSen bereits funktioniert. Konkret wäre dies das openBSD 4.7 Image von Switch. Es wurde die Bash verwendet.

3.2 Abhängigkeiten zum kompilieren der HAPlib

- boost (-iostreams-dev -thread-dev -filesystem-dev -regex-dev werden benötigt)
- graphviz (Achtung: haufenweise Abhängigkeiten!)
- gtk2mm
- swig
- cmake 2.8 (selber bauen, mitgelieferte Version 2.4.3 reicht nicht)
  - für CMake 2.8 wird wiederum der g++ 4.2+ benötigt (lautet dann eg++/egcc)
  - deshalb anpassen:

```
--- cmake-2.8.3-vanilla/bootstrap      Wed Nov  3 20:58:26 2010
+++ cmake-2.8.3/bootstrap        Tue Nov 30 09:00:06 2010
@@ -142,8 +142,8 @@ else
   cmake_default_prefix="/usr/local"
fi

-CMAKE_KNOWN_C_COMPILERS="cc gcc xlc icc tcc"
-CMAKE_KNOWN_CXX_COMPILERS="aCC xlC CC g++ c++ icc como "
+CMAKE_KNOWN_C_COMPILERS="egcc cc gcc xlc icc tcc"
+CMAKE_KNOWN_CXX_COMPILERS="eg++ aCC xlC CC g++ c++ icc como "
CMAKE_KNOWN_MAKE_PROCESSORS="gmake make"

CMAKE_PROBLEMATIC_FILES="\
```

  - und den "neuen" Compiler exportieren:

```
export CXX=eg++
export CC=egcc
```

3.3 Abhängigkeiten von NfSen (sollten schon vorhanden sein)

- apache
- php
- perl + einige Module

3.4 Installation

3.4.1 Entpacken

```
tar xfz hap4nfsen.tar.gz
cd hap4nfsen
```

3.4.2 NfSen

NfSen bitte wie gewohnt installieren, vorzugsweise aber die mitgelieferte Variante(src/nfsen-1.3.5-patched/) wählen. Alternativ müssen einige Dateien gepatcht werden. Die Patches befinden sich unter src/patches/.

3.4.3 HAPlib

```
mkdir build # (Ort egal, solange schreibbar)
cd build
unset CXX #falls noch gesetzt
unset CC #falls noch gesetzt
cmake ../PFADZUHAP4NFSEN -DBUILD_HAP_GUI=false \
-DBUILD_HAP_ENABLEDEBUG=false -DBUILD_HAP_LIB_TEST=false \
-DBUILD_HAP_LIB_HAP4NFSEN=true -DBUILD_HAP_LIB=true \
-DBUILD_HAP=true
make
make install
```

3.4.4 Plugin

```
export BASEPATH=/data/nfsen/ #ggf. anpassen
export HTMLDIR=/var/www/htdocs/nfsen/ #ggf. anpassen
cp -R src/plugins/backend/HAP4NfSen* $BASEPATH/plugins/
cp -R src/plugins/frontend/HAP4NfSen* $HTMLDIR/plugins/
(cd $BASEPATH/plugins/HAP4NfSen/Dot2Graphic/ && ./build.sh)
(cd $BASEPATH/plugins/HAP4NfSen/NfDump2Dot/ && ./build.sh)
```

3.5 Konfiguration

Das Plugin besitzt einige Parameter, welche über die Datei "nfsen.conf" konfiguriert werden können:

Name	Erlaubte Werte	Empfohlener Wert	Beschreibung
max_history	Positive Ganzzahlen	16	Anzahl Schritte eines Benutzers, die in der Session gespeichert werden und mit Hilfe der History Bar wiederhergestellt werden können. Grosse Werte führen zu einem erhöhten Memory-Verbrauch des Frontends. Zudem kann die Darstellung der History Bar unter grossen Werten leiden.
delete_temp_files_after	Positive Ganzzahlen	500	Dauer, nach welcher Dateien im Arbeits- und Bildverzeichnis vom Hap4NfSen Plugin gelöscht werden. Die Aufräumfunktion des Plugins wird vom NfSen Framework periodisch(einmal pro 5 Minuten) aufgerufen. In dieser Funktion werden Dateien in den erwähnten Verzeichnissen, welche älter als der konfigurierte Wert(in Sekunden) sind, gelöscht.
work_dir	Verzeichnispfade	/tmp/HAP4NfSen/	Arbeitsverzeichnis, in dem das Backend Plugin generierte temporäre Dateien ablegt. Falls es nicht bereits existiert, wird das Verzeichnis beim starten des Plugins angelegt. Dazu muss der Benutzer, unter welchem NfSen ausgeführt wird, die entsprechenden Schreibrechte besitzen.
image_dir	Verzeichnispfade, für pic.php erreichbar	/data/nfsen/plugins/	Verzeichnis, in dem generierte Bilder abgelegt werden.

4 Protokolle der Besprechungen

4.1 Woche 1

4.1.1 Datum und Ort

Montag, 20.9.2010, 16:00 - 18:00, Zürich, Switch

4.1.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz
- Peter Haag

4.1.3 Inhalt

- Kennenlernen
- Präsentation NfSen
- Bekanntgabe Eckdaten (Datenmenge, Zeitrahmen, Sitzungen)
- Vorschlag, Testinstallationen durchzuführen und damit spielen
- Netflow von der HSR

4.2 Woche 2

4.2.1 Datum und Ort

Donnerstag, 30.9.2010, 17:30 - 18:30, HSR, 6.112

4.2.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.2.3 Inhalt

- Projekt- und Zeitplanung: Einfach und praktikabel gestalten
- Iteratives Vorgehen verwenden
- Use Cases: Zuerst wenige und einfache definieren, später dann erweitern
- Schwerpunkt der Arbeit liegt bei der Integration in NfSen
- HAP Library: Bestehende Probleme werden von Herrn Glatz genauer untersucht
- Früher Prototyp(Graphik auf Frontend anzeigen) soll schnellstmöglich erstellt werden
- Beide Libraries(HAP und GraphViz) sollen direkt vom Perl Backend aufgerufen werden(keine C++ Standalone Kommandozeilen Applikation verwenden)
- Mögliche Erweiterungen der HAP library:

  - Angabe eines Dateipfades, des spezifiziert, wo die temporäre Datei erstellt werden soll
  - Unterstützung von mehrerern input Files(da jedes File normalerweise Daten für 5 Minuten enthält)

4.2.4 Aufgaben für folgende Woche

- Anpassung des Zeitplans
- Prototyp, der die Libraries(HAP und GraphViz) aus Perl aufruft
- Prototyp "Click Map"

4.3 Woche 3

4.3.1 Datum und Ort

Donnerstag, 7.10.2010, 17:30-18:30, HSR, Cafeteria, Gebäude 1, 1. Stock

4.3.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.3.3 Inhalt

- Präsentation von Prototypen:
  - Einstiegspunkt in NfSen
  - Aufruf der HAP Library aus Perl
  - Umwandlung der erstellten .dot Datei zu SVG mit Perl und Graphviz
- Besprechung verschiedener Möglichkeiten zur Erstellung von click maps(z.B. mit SVG Graphiken)

4.3.4 Aufgaben für folgende Woche

- Graphlet aus NfSen anzeigen
- Liste von zukünftigen Tasks(zur Priorisierung) erstellen

4.4 Woche 4

4.4.1 Datum und Ort

Donnerstag, 14.10.2010, 17:40-18:50, HSR, Cafeteria, Gebäude 1, 1. Stock

4.4.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.4.3 Inhalt

- Unterstützung von IPv6 in HAP: nicht vorhanden
- Präsentation des GUI Prototypen
- Vorschlag: skalierbare SVGs?
- Problem mit anspringen von Sub-Tabs: eventuell Browser bedingt?
- Zum filtern und zusammenführen von Netflow Daten für den Aufruf der HAP Library soll NfDump verwendet werden(Inhalt wird in temporäres File zwischengespeichert)
- Drilldown: Besprechung verschiedener Möglichkeiten
- Arbeitsverzeichnis für NfSen Backend

4.4.4 Aufgaben für folgende Woche

- Erweiterung des GUI Prototypen
  - Soll Daten aus NfSen holen
  - Soll click map besitzen(geklickte IP wird neue Host-IP)

4.5 Woche 5

4.5.1 Datum und Ort

Donnerstag, 21.10.2010, 17:30-17:35, HSR, Cafeteria, Gebäude 1, 1. Stock

4.5.2 Inhalt

- Ausfall, da Herr Glatz Software Engineering im MAS unterrichten musste.
- Nächste Sitzung am Freitag

4.6 Woche 6

4.6.1 Datum und Ort

Freitag, 29.10.2010, 17:30-18:40, HSR, Cafeteria, Gebäude 1, 1. Stock

4.6.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.6.3 Inhalt

- Präsentation des aktuellen Standes
- Fragen zur Darstellung des Graphlet
  - Wunsch von Peter Haag
  - Port zusammenfassung(insbesondere ICMP)
- Extraktion der für den Drilldown benötigten Infos aus .dot Dateien
- Mögliche Termine und Themen für die Besprechung mit Peter Haag

4.6.4 Aufgaben für folgende Woche

- Vorbereitung für Präsentation bei Peter Haag
  - Klickbare Graphkien
  - Fragen für Peter Haag

4.7 Woche 7

4.7.1 Datum und Ort

Montag, 1.11.2010, 16:30-18:00, Zürich, Switch

4.7.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz
- Peter Haag

4.7.3 Inhalt

- Besprechung mit Peter Haag

4.7.4 Aufgaben für folgende Woche

- Vor extrem grossen Graphen warnen
- Wie nach dem desummarisieren wieder summarisieren? Toggle auf Spaltenbeschreibung unten?
- Zoomfunktion erwünscht
- Bei Klick auf Kanten diese highlighten (am besten ganzer Pfad)
- Optional: das Filterfeld auch auf der Pluingseite anzeigen
- Optional: die Bedeutung der Kanten-Texte erklären
- Alle Einsprungspunkte auf der Detailseite berücksichtigen
- Nächstes Meeting: voraussichtlich am 29. November

4.8 Woche 8

4.8.1 Datum und Ort

Donnerstag, 17:20-19:30, HSR, Cafeteria, Gebäude 1, 1. Stock

4.8.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.8.3 Inhalt

- Präsentation des aktuellen Standes
  - Anpassungen am Einstiegspunkt(NfSen Details Seite)
  - History und Undo
  - SVG: Zoom und Pan
- Besprechung von Möglichkeiten zur Erweiterung der HAP lib
  - Möglichkeiten um an die benötigten Daten zu gelangen
  - Verschiedene Typen summarisierter Knoten
  - Das HPG Dateiformat

4.8.4 Aufgaben für folgende Woche

- Erweiterung der HAP lib(Output sollte enthalten, was sich hinter summarisierten Knoten versteckt)
- Erweiterung des Plugins gemäss Liste aus der Besprechung mit Peter Haag

4.9 Woche 9

4.9.1 Datum und Ort

Donnerstag, 18.11.2010, 17:30-19:00, HSR, Cafeteria, Gebäude 1, 1. Stock

4.9.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.9.3 Inhalt

- Präsentation der neuesten Features
  - Erweiterung des Einstiegspunktes(NfSen Details Seite): Zusätzlicher Port Parameter, Unerstützung weiterer NfDump Output Formate
  - Hife Funktion
  - Anzeige der aktuell verwendeten Parameter
  - Erweitertes Highlighting von Graphlet Linen: Neue Farbe und breitere Liniendarstellung
- Besprechung der Bedeutung von ICMP "Ports"
  - Für Filter ignorieren
- Besprechung von Ansätzen zur Erweiterung der .dot Ausgabe der HAP Library
  - Neue Version des HAP Viewer mit aktuallisierter Beschreigung des .hpg Dateiformates
  - Zusätzliche Ausgabe der hinter summarisierten Konten versteckten Daten
- BA: Fortsetzungs des Themas

4.9.4 Aufgaben für folgende Woche

- Weitere Arbeit an Filtering und Drill Down Funktionalität

4.10 Woche 10

4.10.1 Datum und Ort

Mittwoch, 24.11.2010, 14:30-15:00, HSR, Cafeteria, Gebäude 1, 1. Stock

4.10.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.10.3 Inhalt

- Besprechung des Fortschritts
  - Änderungen am Frontend
  - Neues Konzept: NodeId Filter
  - Erweiterung der HAP Lib: .dot Dateien enthalten nun Kommentare mit NodeId Filter Infos

4.10.4 Aufgaben für folgende Woche

- Treffen mit Peter Haag organisieren
  - Abklärung einiger Details
  - Installation des Plugins auf einem SWITCH Server

4.11 Woche 11

4.11.1 Datum und Ort

Freitag, 3.12.2010, 13:30-14:20, Zürich, Switch

4.11.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Peter Haag

4.11.3 Inhalt

- Besprechung des aktuellen Stands
  - Präsentation der Arbeit auf openBSD System
  - Aktuell vorhandene Features
- Verbesserungsvorschläge von Peter Haag
  - Möglichkeit, die automatische Deaktivierung der Zoom und Pan Funktionalität zu umgehen
  - Möglichkeit, die Anzeige von übergrossen charts zuzulassen

4.11.4 Aufgaben für folgende Woche

- Umsetzung der gewünschten Verbesserungen
- Implementation der letzen geplanten Features

4.12 Woche 12

4.12.1 Datum und Ort

Donnerstag, 9.12.2010, 10:00-11:00, HSR, Cafeteria, Gebäude 1, 1. Stock

4.12.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.12.3 Inhalt

- Präsentation und Besprechung aller änderungen seit dem letzten Treffen
  - CMake: Verwendung als build Tool, Anhängigkeiten zu anderen Bibliotheken und Tools
  - LibPcap: Kann beim erstellen der HAP Library für OpenBSD per Parameter auskommentiert weden
  - Überarbeiteter Code in Plugins und HAP Library
  - Graphlet Breite soll für Graphlets ohne JavaScript auf nur 50 Prozent der Breite des Browsers reduziert werden
  - Besprechung von Problemen bei Rollenkonflikten(summarisierter Knoten gehört zu mehreren konkurierenden Rollen)
- NfDump Filter sollen in Zukunft Richtungsinformationen zu Flows enthalten

4.12.4 Aufgaben für folgende Woche

- Implementierung von Richtungsinformationen bei der Filterung mit NfDump
- Per JavaScript eingefärbte linien des Graphen sollen jeweils automatisch in den Fordergrund gebracht werden
- Abschliessende Arbeiten(aufräumen des Codes, abschliessen von Änderungen, Arbeit an Dokumentation und Plakat)

4.13 Woche 13

4.13.1 Datum und Ort

Donnerstag, 16.12.2010, 11:00-12:00, HSR, Cafeteria, Gebäude 1, 1. Stock

4.13.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz

4.13.3 Inhalt

- Besprechung des Fortschritts
  - Z Reihenfolge der Linien des Graphlets(JavaScript Highlighting)
  - Verwendung von Richtungsinformationen für NfDump Filter
  - Erweiterungen in der HAP Lib zur Unterstützung aller Rollentypen
  - Diverse Aufräumarbeiten innerhalb des Codes
  - Erweiterte Browserunterstützung

4.13.4 Aufgaben für folgende Woche

- Fertigstellung der Arbeit
- Erstellung des Plakates bis Montag
- Abgabe der fertigen Arbeit bis Donnerstag, 23.12.2010
- Schlusspräsentation mit Peter Haag planen vorbereiten

4.14 Woche 14

4.14.1 Datum und Ort

Montag, 20.12.2010, 13:00-14:00, ETH Zürich, H83

4.14.2 Anwesende

- Reto Schneider
- Sebastian Hügli
- Eduard Glatz
- Peter Haag

4.14.3 Inhalt

- Präsentation der neuen Features
- Wunsch von Peter Haag: Reduktion von Abhängigkeiten der HAP Library
  - Gewisse Abhängigkeiten wurden bereits bei der Portierung auf OpenBSD entfernt
  - Weitere(z.B. GTK) können entfernt werden, benötigen aber einen Umbau des Codes
- Besprechung von möglichen Erweiterungen und möglichen Inhalten der Bachelorarbeit
  - Diverse Optimierungen des Analysetools(Verbessuerung der Benutzbarkeit, erweiterte Browserunterstützung, ..)
  - Erweiterungen an der HAP Library und dem HAP Viewer
    - Partielle Desummarisierung von Knoten
    - Erweiterte Auswertung von Rollen, neue Auflösung von Konflikten
    - Möglichkeit, Aktivität in verschiedenen Zeitfenstern zu vergleichen
    - Unterstützung von IPv6
  - Mehr Übersicht und ein erweiterter Einstiegspunkt ins Plugin durch die Integration von Afterglow(eine weitere Visualisierungssoftware)

5 Rückblicke

5.1 Reto Schneider

5.1.1 Projektverlauf

Da wir weder eine eigene Arbeit vorgeschlagen, noch von einer vorgeschlagenen Arbeit restlos begeistert gewesen waren, hatten wir uns bei Herr Glatz gemeldet um mehr über den HAPviewer zu erfahren. Nach einer Demo des HAPviewers an der ETH war für uns sofort klar, dass wir dieses Projekt umsetzen wollen. Bereits in der ersten Woche trafen wir uns zusammen mit Peter Haag, Mitarbeiter bei SWITCH, und tauschten uns aus. In den darauf folgenden Wochen machten wir uns mit den Technologien (HAPviewer, NfSen, NfDump, Perl, PHP, z.T. OpenBSD, Latex, C/Perl Bindings, etc) vertraut und erstellten einen rudimentären Prototypen. Mit diesem wurden wir nach gut sieben Wochen wiederum bei SWITCH vorstellig, wo wir einiges an Feedback und neue Ideen erhielten. Danach entwickelten wir dann die (für diese Arbeit) finale Version, welche wir am 20. Dezember wiederum Peter Haag präsentierten und auch gleich einige Ideen für die kommende Bachelorarbeit austauschten.

5.1.2 Rückblick

Das Projekt verlief in sehr geordneten Bahnen, grössere Unfälle gabe es keine. Sehr hilfreich war, dass Herr Eduard Glatz als auch Peter Haag sehr viel Zeit für unser Projekt aufbrachten und jederzeit für Fragen zur Verfügung standen. Gestört hat mich, dass ich lange Zeit nicht realisiert habe, dass der Ansatz, alle benötigten Informationen aus dem .dot-File auszulesen sehr viel unnötigen Aufwand mit sich brachte, und wir erst, nachdem wir viele Stunden in die Auswertung der .dot-Datei gesteckt hatten, zur derzeitigen Lösung (Erweiterung der HAP-Bibliothek) umgeschwenkt sind.
Ebenfalls stört mich, dass die Erweiterung der HAP-Bibliothek alles andere als schön ist (was hoffentlich wärend der Bachelorarbeit korrigiert werden kann).

5.1.3 Gelerntes

Während dem Projektverlauf konnte ich so einiges über Perl und OpenBSD, beides Neuland für mich, sowie, in kleinerem Umfang, auch bezüglich PHP, CMake und Latex lernen.

5.1.4 Fazit

Das Wichtigste voraus: Es freut mich sehr, dass sich Sebastian als ein wirklich toller Teamkollege herausgestellt hat - vor der Semesterarbeit kannte wir uns nochnicht, was natürlich ein gewisses Risiko mit sich brachte. Das Projekt war für mich eine sehr positive Erfahrung. Die Arbeit war interessant und wir konnten die Software in einen, zwar nicht perfekten, aber durchaus benutzbaren Zustand bringen. Ich für meinen Teil freue mich sehr, zusammen mit allen Beteiligten im nächsten Semester mit der Arbeit weiter zu fahren.

5.2 Sebastian Hügli

5.2.1 Projektverlauf

Gleich als diese Arbeit ausgeschrieben wurde, hat sie uns angesprochen. Wir erhielten dann kurz darauf die Möglichkeit, den HAPviewer, welcher zu dieser Zeit noch nicht öffentlich verfügbar war, bei Herrn Glatz in der ETH zu begutachten. Danach waren wir gleich von der Arbeit überzeugt. Um das Semester gut ausnutzen zu können, begannen wir bereits vor dem Beginn der Studienarbeit, uns mit den beiden Tools vertraut zu machen.
In den ersten Tagen hatten wir bereits die Möglichkeit, uns zum ersten Mal mit Peter Haag vom Industriepartner SWITCH zu treffen. Dadurch erhielten wir ersten input und konnten unser Verständnis für das NfSen Framework verbessern.
In den folgenden Wochen arbeiteten wir dann an den ersten Prototypen, welche die Grundfunktionalitäten wie etwa den Aufruf von C Code aus Perl oder das erweitern von Graphiken um Image Maps enthielten. Danach fügten wir alle einzelnen Prototypen zu einem ersten Plugin zusammen.
Diese erste Version des Plugins durften wir nochmals bei SWITCH präsentieren und konnten in einer darauf folgenden Brainstorming Session viele Ideen für erweiterungen sammeln. In den folgenden Wochen haben wir unser Plugin dann um viele zusätzliche Features erweitert. Dazwischen zeigten wir den Fortschritt Herrn Glatz wöchentlich, und erhielten dabei hilfreiches Feedback.
Einige Wochen vor Abschluss der Arbeit portierten wir unsere Arbeit noch auf OpenBSD und durften sie nochmals bei SWITCH vorzeigen. Somit erhielten wir nochmals hilfreichen Input für den Abschluss des Projekts.
Am Montag der letzten Woche trafen sich alle Involvierten nochmals zu einer Schlusspräsentation an der ETH, wobei ebenfalls über die Intagration des Plugins in die nächste Version von NfSen, sowie mögliche erweiterungen gesprochen wurde.

5.2.2 Rückblick

Insgesammt bin ich mit dem Verlauf, sowie dem Resultat der Arbeit sehr zufrieden. Allerdings gibt es einige Punkte, welche besser gelöst werden könnten. Einer davon ist die Abschätzung des Aufwands verschiedener Arbeiten. Es gelang uns schon sehr früh, einen einfachen Prototypen zu erstellen, doch der Umgang mit den Filtern war aufwändiger als zuerst erwartet. Zu Anfang der Arbeit nahmen wir an, das alle Informationen, welche für NfDump Filter benötigt werden, unkompliziert aus der erstellten .dot Datei gelesen werden können. Am Ende musste die HAP Library erweitert werden, um dies zu ermöglichen.
Etwas weiteres war, das wir zu Anfang des Projektes zwar die Struktur der Dokumentation erstellt haben, während des laufenden Projektes aber nicht regelmässig den aktuellen Fortschritt nachtrugen.

5.2.3 Gelerntes

Zu Beginn der der Arbeit kannte ich mit vielen der eingesetzten Technologien wie Perl oder PHP gar nicht aus. Im Laufe dieser Arbeit hatte ich die Möglichkeit, mich darin einzuarbeiten, so das die Arbeit damit inzwischen leicht fällt.
Daneben habe ich zur Erstellung der Dokumentation zum ersten Mal mit LaTeX gearbeitet, was mir zu Anfang einige Schwierigkeiten bereitete. Nach den ersten

Wochen, in denen die Struktur des Dokuments erstellt wurde, und der damit gewonnenen Erfahrung ging es immer leichter. Inzwischen finde ich die Arbeit mit LaTeX ziemlich angenehm und plane es für die Bachelorarbeit ebenfalls einzusetzen.
Auch im Projekt Management konnte ich einiges dazulernen. Durch regelmässige Treffen mit Betreuer und Industriepartner konnten wir uns von einem anfangs einfachen Prototypen Schritt für Schritt zum endgültigen Produkt vorarbeiten und Probleme frühzeitig erkennen.
Zudem war es durch den Umfang der Studienarbeit gut möglich, bestimmte Aufgaben aufzuteilen, was dazu führte, dass wir uns genau miteinander absprechen und Schnittstellen definieren mussten.

5.2.4 Fazit

Die Arbeit mit NfSen und HAPviewer war sehr interessant und abwechslungsreich. Besonders gut gefallen hat mir dabei die zu Anfang des Projeks noch ziemlich offene Aufgabenstellung. Dadurch war es möglich, uns Schritt für Schritt zu einem Plugin vorzuarbeiten und dabei auch eigene Ideen einzubringen.
Sehr erfreulich ist ebenfalls, das unser Industriepartner mit dem Resultat zufrieden ist und das Plugin in das NfSen Basis-Framework übernehmen will. Im nächsten Semester werden wir als Bachelorarbeit eine Folgearbeit zu einem ähnlichen Gebiet schreiben, worauf ich mich freue. Durch diese Arbeit konnte ich in diesem Themenbereich viel neues dazulernen und denke, das ich für die bevorstehende Bachelorarbeit gut vorbereitet bin.

Merkblatt: Umgang mit vertraulichen Daten

Dieses Merkblatt beinhaltet Beispiele und relevante Informationen (Gesetzes Artikel) zum Umgang mit vertraulichen Daten. Die untenstehenden Auflistungen sind nicht abschliessend.

Verhalten in der Öffentlichkeit (Pausen, Verkehrsmitteln)

In Pausen, beim Mittagessen, im Zug muss darauf geachtet werden, dass vertrauliche Informationen nicht von Dritten mitgehört oder eingesehen werden können. Das Druck-, Surf- und E-Mailverhalten von HSR Angehörigen ist streng vertraulich und darf nicht öffentlich diskutiert werden.

Speicherung und Übertragung von Daten

Es dürfen keine permanenten Kopien von vertraulichen Daten auf dem persönlichen Notebook, Memorystick oder andern mobilen Datenträger bestehen. Die Daten auf Datenträgern welche vor allem zur Übertragung dienen, müssen nach der Übertragung gelöscht werden.

Wechsel der Abteilung

Bei einem Wechsel in eine andere Abteilung ist es wahrscheinlich, dass immer noch Berechtigungen für den Zugang zu Systemen oder Daten der alten Abteilung bestehen. Solche Berechtigungen sind der zuständigen Stelle zu melden damit die Berechtigungen entfernt werden können. Ein Anrecht auf Berechtigungen besteht nicht implizit.

Mehrere Rollen gleichzeitig / Rollenwechsel

Bei verschiedenen gleichzeitigen Rollen, zum Beispiel Student und Mitarbeiter, muss beachtet werden, dass die Kompetenzen der Mitarbeiterrolle in der Ausübung der Studentenrolle nicht verwendet werden. Einblick in Zeugnisse, Dateien oder Informationsbeschaffung über Studierende sind Beispiele solcher Kompetenzüberschreitungen.

Umgang mit Berechtigungen

Personendaten, Finanzdaten, E-Mails oder andere vertrauliche Informationen aus Datenbanken oder Datensammlungen dürfen nicht an Dritte weitergegeben werden. Über den Zugang zu vertraulichen Daten dürfen keine persönlichen Vorteile verschafft werden.

Zugang zu persönlichen Daten

Beim Zugang zu Systemen (PC, Laptop) von anderen Personen, zum Beispiel zu Wartungszwecken, dürfen nur die Bereiche und Daten beachtet werden, welche für die Ausübung der Tätigkeit relevant sind. Der Verlockung interessante Dateien zu öffnen ist strikte zu widerstehen.

Zugang zu Arbeitsräumen

Beim Zugang zu fremden Büros ist das Büro so zu verlassen, wie es angetroffen wurde. Zudem muss kommuniziert werden, dass man den Raum betreten hat und welche Tätigkeiten ausgeführt wurden. Herumliegende Dokumente dürfen nicht beachtet werden.

Wartung- und Testarbeiten

Werden vertrauliche Daten für Tests oder Wartung verwendet, darf nur mit den eigenen Daten getestet werden (z.B. Dynamics, SAS, Badge, Uniflow, SiPass). Ist dies nicht möglich, muss das Einverständnis der Datenbesitzer eingeholt werden.

Art. 35 des Datenschutzgesetzes

Art. 35 Verletzung der beruflichen Schweigepflicht

Wer vorsätzlich geheime, besonders schützenswerte Personendaten oder Persönlichkeitsprofile unbefugt bekanntgibt, von denen er bei der Ausübung seines Berufes, der die Kenntnis solcher Daten erfordert, erfahren hat, wird auf Antrag mit Haft oder mit Busse bestraft. Gleich wird bestraft, wer vorsätzlich geheime, besonders schützenswerte Personendaten oder Persönlichkeitsprofile unbefugt bekanntgibt, von denen er bei der Tätigkeit für den Geheimhaltungspflichtigen oder während der Ausbildung bei diesem

6 Glossar

Begriff	**Beschreibung**
AJAX	AJAX(Asynchronous JavaScript and XML) ist ein Konzept, welches asynchronen Datenaustausch zwischen Server und Webbrowser erlaubt.
C	Eine weit verbreitete prozedurale Programmiersprache.
C++	Eine Programmiersprache mit prozeduralen und objektorientierten Elementen.
CMake	Ein Buildsystem
Graphlet	Eine vom HAP Viewer generierter Graph, welcher aus 5 Partitionen(Host IP, Protokoll, Host Port, Destination Port und Destination IP) besteht.
GraphViz	GraphViz(Graph Visualization Software) ist eine Software, welche aus Graphenbeschreibungen Bilder generiert.
HAPviewer	Der HAPviewer(Host Application Profile Graphlet Viewer) ist eine Applikation zur Analyse von Netzwerkverbindungen.
HAP4NfSen	Das im Rahmen dieser Arbeit erstellte Plugin für NfSen
"hpg"	**H**ost **P**rofile **G**raphlet - Ein properitäres Format von Prof. Eduard Glatz um Graphen platzsparend zu speichern. Wird vom HAPviewer intern verwendet.
HTML	HTML(HyperText Markup Language) ist eine Sprache, die die Darstellung von Webseiten beschreibt.
JavaScript	Die Skriptsprache Java Script wird hauptsächlich innerhalb von Web Browsern verwendet, um Webseiten dynamischer zu gestalten.
Netflow	Ein Tupel von Verbindungsdaten, das Informationen wie die von beiden Seiten verwendeten Ports enthält.
NfDump	NfDump(Netflow Dump) ist eine Kommandozeilen Applikation, mit welcher man Netflow Dateien Filtern sowie statistische Auswertungen erstellen kann.
NfSen	NfSen(Netflow Sensor) ist eine Applikation zum sammeln und analysieren von Netflows

Perl	Eine Skriptsprache.
PHP	PHP(PHP: Hypertext Preprocessor) ist eine Skriptsprache welche zur erzeugung von dynamischen Webseiten verwendet werden kann.
SVG	SVG(Scalable Vector Graphics) ist ein XML basiertes Format für Vektorgraphiken.
SVGPan	Eine JavaScript Library mit der eine SVG Graphik um Zoom und Pan Fumktionalität erweitert werden kann.
SWIG	**S**implified **W**rapper and **I**nterface **G**enerator) ist ein Tool, um im C/C++ geschriebene Programme in anderen Programmiersprachen (in unserem Fall Perl) verfügbar zu machen.

List of Figures

References

[1] nfdump.sourceforge.net/, 24.10.2010

[2] http://hapviewer.sourceforge.net/, 1.12.2010

[3] http://nfdump.sourceforge.net/overview.gif, 1.10.2010

[4] http://www.ripe.net/ripe/meetings/ripe-50/presentations/ripe50-plenary-tue-nfsen-nfdump.pdf, s. 14, 1.10.2010

[5] http://nfsen.sourceforge.net/PluginGuide/plugin-guide.html, 17.10.2010