

Internetportal für Computernetze-Praktika

Diplomarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Stefan Zimmerli
2002

Leiter der Arbeit:
Prof. Dr. Torsten Braun

Forschungsgruppe Rechnernetze und Verteilte Systeme (RVS)
Institut für Informatik und angewandte Mathematik

Leiter der Arbeit:
Prof. Dr. Torsten Braun

Forschungsgruppe Rechnernetze und Verteilte Systeme (RVS)
Institut für Informatik und angewandte Mathematik

Betreuer der Arbeit:
Marc-Alain Steinemann

Forschungsgruppe Rechnernetze und Verteilte Systeme (RVS)
Institut für Informatik und angewandte Mathematik

Zusammenfassung

Im Rahmen des "Swiss Virtual Campus" (SVC) wird ein Projekt namens "Virtual Internet Telecommunications Laboratory of Switzerland" (VITELS) durchgeführt. An diesem Projekt sind mehrere Universitäten und Fachhochschulen beteiligt, mit dem Ziel ein gemeinsames verteiltes Fernkurslabor im Bereich der Rechnernetze aufzubauen. Anders als bei herkömmlichen Praktikumsarbeiten sollen die StudentInnen die Möglichkeit haben, von zu Hause aus und zu beliebiger Zeit auf das Fernkurslabor zuzugreifen. Die einzelnen Kursmodule werden derzeit an den verschiedenen Hochschulen im Rahmen des Projekts VITELS entwickelt.

Das Fernkursmodul IP Security der Universität Bern basiert auf einem im Rahmen des „Praktikums Computernetze“ angebotenen Versuchs, bei dem mittels zwei Cisco Routern ein Virtuelles Privates Netzwerk aufgebaut wird und Messungen durchgeführt werden. Ziel dieser Arbeit war es, diesen traditionellen Praktikumsversuch mittels eines Internetportals StudentInnen für den Fernzugriff anzubieten.

In einer ersten Phase wurde daher ein Lösungskonzept für das Internetportal erarbeitet, welches besonderen Augenmerk auf Sicherheit und Robustheit legt, sei es um den Zugriff auf das Fernkursmodul nur registrierten Benutzern zu gewähren oder die Geräte im Falle einer Fehlkonfiguration wieder in einen definierten Zustand zu bringen.

Die in der Diplomarbeit „Authentication, Authorization and Resource Reservation for Distributed Laboratories“ vorgeschlagene und zusammen umgesetzte Architektur für das VITELS-Projekt umfasst ein leistungsfähiges Reservierungssystem und eine Studentendatenbank auf der Basis des Lightweight Directory Access Protocols (LDAP).

In dieser Arbeit wird beschrieben, wie mittels des konzipierten und implementierten Internetportals die vorhandene Hardware, wie Cisco Router und Linux-Rechner an das Reservierungssystem angeschlossen wird und in die VITELS-Architektur eingebunden wird. Des weitern wurde ein Mechanismus entwickelt, um ein gesetztes Administrationspasswort eines Routers wieder zurück zu setzen und mittels einer in PHP geschriebenen Benutzerschnittstelle die Geräte von den StudentInnen über einen normalen Browser fernkonfiguriert werden können.

Danksagung

Ich möchte an dieser Stelle allen Personen danken, die mich während der Dauer meiner Diplomarbeit unterstützt haben. Mein ganz besonderer Dank gilt Prof. Dr. T. Braun, der es mir erlaubte, meine Diplomarbeit in seiner Forschungsgruppe durchzuführen. Weiter spreche ich meinen speziellen Dank an Marc-Alain Steinemann aus, der mich während dieser Arbeit betreute und der sich immer Zeit für mich nahm, um Ideen und Probleme zu diskutieren. Er investierte viel Zeit um meine Arbeit zu begutachten und um hilfreiche Vorschläge und Korrekturen anzubringen. Danken möchte ich auch Matthias Scheidegger, der mir mit seinen PERL-Kenntnissen oftmals mit Rat zur Seite stand. Weiterer Dank geht an Roland Trummer von den Informatikdiensten der Universität Bern, der mir bei der Anbindung des Portals an das produktive Kurs- und Reservierungssystem und bei der Problembhebung eine grosse Hilfe war.

Ganz besonderer Dank gebührt meinen Eltern, ohne deren Unterstützung ich das Informatikstudium nicht hätte durchführen können.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Gesetzte Ziele.....	3
1.3	Erreichte Ziele.....	3
1.4	Übersicht.....	4
2	Verwandte Arbeiten.....	5
2.1	Verteiltes Lernen.....	5
2.1.1	Definition.....	5
2.1.2	Ziele.....	5
2.1.3	Anforderungen.....	6
2.2	Übergeordnetes Programm: Swiss Virtual Campus (SVC).....	7
2.3	Beispiele von Internet-basierten Kursen, Kursverwaltungssystemen und Werkzeugen.....	8
2.3.1	Einfache Webkurse: W3 Schools.....	8
2.3.2	Kursverwaltungssysteme.....	8
2.3.2.1	WebCT.....	8
2.3.2.2	Top Class.....	8
2.3.2.3	Lotus Learning Space.....	8
2.3.3	Entfernte Experimentierplattformen.....	9
2.3.3.1	Mentortech.....	9
2.3.3.2	Emulab.....	9
2.3.3.3	Nano-World.....	9
2.3.4	Werkzeuge für verteiltes Lehren: ARIADNE.....	10
3	Grundlegende Techniken.....	11
3.1	Betriebssysteme und Server.....	11
3.1.1	Linux.....	11
3.1.2	Apache.....	11
3.1.3	Lightweight Directory Access Protocol (LDAP).....	11
3.2	Programmiersprachen.....	12
3.2.1	PHP.....	12
3.2.2	Perl.....	12
3.3	Sicherheitsrelevante Protokolle.....	13
3.3.1	Secure sockets Layer (SSL) / Transport Layer Security (TLS).....	13
3.3.2	Secure Shell (SSH).....	13
3.3.3	Virtuelle Private Netzwerke (VPN).....	14
4	Das traditionelle Modul „IPSec“ des Computernetze-Praktikums.....	16
4.1	Ziel.....	16
4.2	Ablauf.....	17
4.3	Aufbau.....	17
4.4	Konfiguration der Linux-Rechner.....	18
4.4.1	Netzwerkkarten-Konfiguration mit ifconfig.....	18
4.4.2	IP-Routing-Konfiguration mit route.....	18
4.5	Konfiguration der Cisco Router.....	19
4.5.1	Zugriff über Netzwerk / Konsole.....	19
4.5.2	Minicom.....	19
4.5.3	Netzwerk-Konfiguration der Router.....	20
4.6	Testen der Konfiguration mit Ping und Traceroute.....	22

4.7	Bandbreitenmessung mit Netpipe	23
4.8	Netzwerk-Sniffing mit Tcpdump	23
4.9	Konfiguration des VPN-Tunnels zwischen den Routern	24
4.10	Bandbreitenmessung mit und ohne VPN-Tunnel	27
4.11	Mitlesen (“Sniffen”) von Passwörtern mit und ohne VPN-Tunnel	29
5	Die VITELS-Architektur	32
5.1	Kursserver	34
5.2	Studentendatenbank	36
5.2.1	Verzeichnisstruktur für Universitäten	36
5.2.2	Verzeichnisstruktur für VITELS-spezifische Einträge	37
5.3	Reservierungssystem	38
5.3.1	Attribute der Timetable-Einträge	39
5.3.2	Attribute der Module-Einträge	39
6	Internetportal	41
6.1	Anforderungen an das Internetportal	41
6.2	Das Konzept des Internetportals	43
6.2.1	Das Portal als Hub	43
6.2.2	Das Portal als Firewall	44
6.2.3	Abbildung der Portalbenutzer auf Laborgeräte	44
6.2.4	Authentifizierungsmethoden	44
6.2.4.1	Authentifizierung mittels lokalen Benutzer- / Passwortlisten	44
6.2.4.2	Authentifizierung mittels des Network Information Service (NIS)	45
6.2.4.3	Authentifizierung mittels LDAP	45
6.3	Realisierung	46
6.3.1	Ablauf der Anmeldung	46
6.3.2	Benutzerschnittstelle (Webfrontend)	48
6.3.3	Bildschirmabzüge der Benutzerschnittstelle und des SSH-Applets	49
7	Implementierung des Internetportals für das Fernkursmodul IPsec	55
7.1	Unterschiede zwischen der allgemeinen Realisierung des Internetportals und der Implementierung für das Fernkursmodul IPsec	56
7.1.1	Anmeldeweiterleitung an Linux-Rechner bzw. Cisco Router	56
7.1.2	Integration des Rechners host2 in das Portal	57
7.2	Automatisches Löschen eines gesetzten Cisco Router Administrationspassworts	60
7.2.1	Ansatz 1: Der Serial-Line Sniffer	60
7.2.2	Ansatz 2: Die Cisco Password Recovery Solution	61
7.2.2.1	Manuelles Vorgehen	61
7.2.2.2	Die Relaiskarte	64
7.2.2.3	Der implementierte Relaiskartentreiber	65
7.2.2.4	Die implementierte Routerbibliothek	66
7.2.2.5	Das implementierte Reset-Skript	67
8	Zusammenfassung und Ausblick	68
8.1	Resultate	68
8.2	Diskussion	68
8.3	Ausblick	69
8.3.1	Auswahl des Router IOS	69
8.3.2	Zusätzliche Konfiguration der Linux-Rechner	69
8.3.3	Automatisieren der Kontrolle der Messresultate	69
8.3.4	Anpassen des Internetportals für andere Geräte	70
9	Anhang	71

9.1	Implementierter Quellcode	71
9.1.1	Die Relaiskartentreiberbibliothek RelaisLib.pm	71
9.1.2	Die Routerbibliothek RouterLib.pm	76
9.1.3	Das Routerresetskript pw_reset.pl	80
9.2	Referenzen	82
9.3	Abbildungsverzeichnis.....	85
9.4	Tabellenverzeichnis	87
9.5	Verzeichnis der Konfigurations- und der Quellcodedateien.....	88

1 Einleitung

1.1 Motivation

Das Internet hat sich in den letzten Jahren zu einem bedeutenden Medium für den Informationsaustausch zwischen Menschen auf der ganzen Welt entwickelt. Waren es bis Mitte der neunziger Jahre vor allem Hochschulen und Universitäten, die vom Internet Gebrauch machten, steht mittlerweile beinahe in jedem Haushalt ein Rechner, der mit dem Internet verbunden werden kann, zur Verfügung. Die am meisten genutzten Dienste des Internets sind ohne Zweifel die elektronische Post (Email), die es ermöglicht, in wenigen Sekunden eine Nachricht rund um den Globus zu schicken, sowie die schier unendlich erscheinende Sammlung von publizierten Informationen im World Wide Web. Das Internet erscheint daher eine optimale Möglichkeit, der Mobilität des Menschen gerecht zu werden und es ihm zu erlauben, unwichtig wo er sich befindet, zu beliebiger Zeit seinem Lern- und Wissensdurst zu stillen.

Doch gibt es auf dem Weg vom bisherigen klassischen Unterricht zum Fernunterricht verschiedene Hürden zu überwinden, da zwischen den beiden Unterrichtsformen markante Unterschiede bestehen. Beim klassischen Unterricht ist es für die Schüler jederzeit möglich, dem Lehrer Fragen zu stellen oder eine Bemerkung einzubringen, während er beim Fernunterricht sich alleine mit der Materie befassen muss und Fragen zum Beispiel per Email an den Lehrer richten muss. Auch profitiert man im Klassenverband von den Fragen oder Bemerkungen der Mitschüler und kann Unklarheiten in der Pause diskutieren. Des weiteren müssen die Unterrichtsmittel für den Fernunterricht angepasst werden, denn es reicht nicht, das Manuskript mit dem Lernstoff vom Papier auf eine Videokassette oder eine Web-Site zu übertragen oder eine Präsentation, die normalerweise vor einem Plenum vorgetragen wird, im Internet zu publizieren.

Das schweizerische Bundesamt für Bildung und Wissenschaft [1] hat die obgenannten Probleme erkannt und ein Programm namens „Swiss Virtual Campus“ (SVC) [2] lanciert. Ziel dieses Programms ist es, Lehrmaterial für den Internet-basierten Unterricht an Universitäten zu schaffen. Es soll qualitativ hochwertiges Lehrmaterial entwickelt werden. Die Kurse sollen nach Möglichkeit mehrsprachig sein und modernen didaktischen sowie ergonomischen Richtlinien genügen. Weiter sollen die Beziehungen zwischen den Universitäten vertieft werden. Dazu müssen alle Projekte des SVC aus mehreren Partnern aus den Bereichen Hochschulen und Industrie bestehen.

Es wurden 50 Projekte akzeptiert. Jedes Projekt entwickelt einen Kurs, welcher übers Internet besucht werden kann und Lehrmaterial, Übungen, Seminare oder praktische Arbeiten, sowie Online-Hilfe beinhaltet.

Eines dieser Projekte ist das „Virtual Internet and Telecommunications Laboratory of Switzerland“ (VITELS) [3]. Ziel dieses Projektes ist es, ein virtuelles Computer Netzwerk Labor mit mehreren Kursmodulen übers Internet zugänglich zu machen. StudentInnen können mittels eines normalen Browsers auf VITELS zugreifen. Anders als in einem traditionellen Labor, soll die StudentInnen von ihren eigenen

Rechnern aus die vorhandene Hardware konfigurieren und Messungen durchführen können ohne hierzu weitere Programme auf den Rechnern installieren zu müssen.

Die einzelnen Kursmodule (siehe auch Tabelle 1) werden von verschiedenen Institutionen (Universitäten Bern, Fribourg, Genf, Neuenburg und der Fachhochschule Fribourg) entwickelt und gepflegt und zusammen in ein einheitliches webbasiertes Kurssystem integriert. Als Kursverwaltungssystem wurde WebCT [4] ausgewählt, welches im Kapitel 2.3.2.1 näher beschrieben wird. Die Entwicklung durch mehrere Institutionen ermöglicht es, dass die Benutzer die vorhandenen Ressourcen an geografisch verteilten Orten benutzen und von den Erfahrungen und dem Wissen der teilnehmenden Organisationen profitieren können.

Ein wichtiger Punkt ist der Zugriff auf die Ressourcen der Kursmodule. Nur autorisierte Teilnehmer sollen diese benutzen können. Bisher pflegte jede Institution ihre eigene Studentendatenbank und es musste somit eine Möglichkeit gefunden werden, entweder die vorhandenen Daten abgleichen zu können oder eine zusätzliche zentrale Datenbank aufzubauen, gegen welche die Teilnehmer authentifiziert werden. Hinzukommend musste sicher gestellt werden, dass nicht mehrere StudentInnen gleichzeitig auf die Ressourcen zugreifen können, da diese (zum Beispiel Router oder Rechner) nur in begrenzter Anzahl verfügbar sind. Weiter mussten Sicherheitsmassnahmen getroffen werden, um die sensiblen Daten der StudentInnen wie Benutzernamen und Passwörter gesichert zwischen den teilnehmenden Instituten auszutauschen.

Aus diesen Gründen wurde zusammen mit der Diplomarbeit „Authentication, Authorization and Resource Reservation for Distributed Laboratories“ [5] eine Architektur implementiert, die obigen Ansprüchen gerecht wird. Zwischen den Rechnern werden nur gesicherte Verbindungen hergestellt. Ausserdem wurde ein Verwaltungssystem entwickelt, mittels dessen StudentInnen die Ressourcen für eine bestimmte Zeit reservieren können und damit exklusiven Zugriff auf die Hardware haben.

In dieser Diplomarbeit wird beschrieben, wie am Institut für Informatik und angewandte Mathematik (IAM) [6] der Universität Bern [7] vorhandene Hardware wie Cisco Router [8] und Linux-Rechner eines traditionellen Labors mittels eines Portals StudentInnen für den Fernzugriff zugänglich gemacht wurden und in eine verteilte Lernarchitektur integriert wurden. Die so integrierte Hardware kann als Modul „IP Security“ im Rahmen des VITELS-Kurses mittels eines Webbrowser benutzt und konfiguriert werden.

Kursmodule des VITELS-Basiskurses
Linux Installation und Konfiguration
Simulation der IP Netzwerk Konfiguration
Konfiguration und Messung eines realen IP Netzwerks
Client- / Serverprogrammierung
Protokoll Analyse
IP Security
Firewallverwaltung

Tabelle 1: Kursmodule des VITELS-Basiskurses

1.2 Gesetzte Ziele

Die für diese Diplomarbeit gesetzten Ziele wurden durch die beiden folgenden Teilbereiche festgelegt:

- 1) Um die für den Kurs reservierten Cisco Router fern konfigurieren zu können, muss ein Portal konzipiert werden, welches Konfigurationsanweisungen der Benutzer, die über ein Webinterface eingegeben werden, entgegen nimmt und an den entsprechenden Router weiterleitet. Dieser führt die Anweisungen aus und schickt seine Meldungen via Portal an den Benutzer zurück. Dieses muss exemplarisch für den im Rahmen des Praktikums „Computernetze“ vorhandenen IPSec-Versuch implementiert werden. Dabei müssen auch Linux-Rechner mittels des Portals für den Fernzugriff zugänglich gemacht werden. Schliesslich muss das Portal an das am IAM entwickelte LDAP-basierte Reservierungssystem [5] angebunden werden.
- 2) Implementierung einer Reset-Funktion, welche es bei Bedarf ermöglicht, die Cisco Router in ihre Ausgangskonfiguration zurückzusetzen. Mittels dieser Funktion sollen die StudentInnen im Falle einer Fehlkonfiguration die Router in einen definierten Ausgangszustand bringen können, um danach eine Neukonfiguration vorzunehmen. Zusätzlich soll durch die Funktion ein gesetztes Administrationspasswort der Router gelöscht werden. Das Auslösen dieser Reset-Funktion soll den Studenten über das Webinterface des Portals angeboten werden.

1.3 Erreichte Ziele

Die folgenden beiden Aspekte wurden im Rahmen dieser Diplomarbeit erreicht:

- 1) Das Konzept des Portals wurde gemäss den Anforderungen erstellt und exemplarisch implementiert. Es ermöglicht registrierten StudentInnen, die vorhandenen Cisco Router über ein Webinterface zu konfigurieren, sowie auf Linux-Maschinen zuzugreifen, um damit Netzwerkmessungen durchzuführen. Zusätzlich wurde das Portal an ein Reservierungssystem angebunden, welches sicherstellt, dass jeweils nur ein Benutzer im Fernkurslabor aktiv sein kann.
- 2) Die Reset-Funktion wurde gemäss den Vorgaben implementiert und ermöglicht es, via Webinterface die Cisco Router in ihre Ausgangskonfiguration zurück zu setzen. Damit wird einerseits ein gesetztes Administrationspasswort auf den Routern gelöscht und andererseits die Konfiguration in einen definierten Ausgangszustand gebracht, in welchem die Router unkonfiguriert sind. So mit ist es den StudentInnen ohne weiteres möglich, eine Neukonfiguration der Router vorzunehmen.

1.4 Übersicht

Die Struktur dieses Dokuments ist wie folgt: Im zweiten Kapitel wird vertieft auf das Thema „Verteiltes Lernen“ eingegangen und es werden Ziele und Anforderungen an verteilte Lehrsysteme beschrieben. Des Weiteren wird das Projekt Swiss Virtual Campus vorgestellt und Beispiele für andere existierende verteilte Lehrsysteme angegeben. Im dritten Kapitel werden die für die Realisierung dieser Arbeit benötigten grundlegenden Techniken vorgestellt und erklärt. Kapitel 4 beschreibt das traditionelle Computernetze-Praktikumsmodul IPSecurity. Im fünften Kapitel wird auf die VITELS-Architektur und deren Komponenten eingegangen. Im sechsten Kapitel werden die Anforderungen an das Internetportal beschrieben, dessen Konzept und die allgemeine Realisierung erläutert. Das Kapitel 7 beschreibt die exemplarische Umsetzung des Portalkonzeptes und die notwendigen Implementierungen, um das traditionelle Praktikumsmodul IPsec als VITELS-Fernkursmodul anbieten zu können. Im Kapitel 8 werden die gewonnenen Resultate diskutiert und ein Ausblick für Erweiterungen des Internportals gegeben, bevor die Arbeit durch den Anhang beschlossen wird.

2 Verwandte Arbeiten

2.1 Verteiltes Lernen

2.1.1 Definition

Verteiltes Lernen (englisch Distance learning, [12]) wird dadurch definiert, dass der Dozent und seine Zuhörer bzw. die Unterrichtsinhalte und der Student örtlich voneinander getrennt sind.

Elektronisches Lernen (englisch e-Learning, [12]) zeichnet sich dadurch aus, dass ausserhalb eines Klassenzimmers unter Benutzung der Internet-Technologie gelehrt und gelernt wird.

Des weiteren kann Verteiltes Lernen in die zwei Bereiche **synchrones verteiltes Lernen** (bei dem der Dozent und seine Zuhörer in Echtzeit und interaktiv miteinander kommunizieren) und **asynchrones verteiltes Lernen** (bei dem der Dozent den Kurs vorgängig erstellt hat und die Teilnehmer diesen zu beliebiger Zeit besuchen) unterteilt werden.

2.1.2 Ziele

Die Ziele, die zur Realisierung von verteilten Lernsystem führen, sind vielfältig und können wie folgt genannt werden:

- **Ortsunabhängigkeit**
- **Zeitunabhängigkeit**
- **Kostenoptimierung**
- **Bessere Auslastung von Ressourcen**

2.1.3 Anforderungen

Um die in Kapitel 2.1.2 genannten Ziele zu erreichen, müssen verschiedene Anforderungen an das verteilte Lehrsystem erfüllt werden. Die wichtigsten dieser Anforderungen werden im folgenden genannt:

- **Registrierung**
Ein verteiltes Lehrsystem muss so konfiguriert werden können, dass nur registrierte Personen darauf zugreifen können und diese eindeutig identifiziert (z.B. mittels Benutzernamen/Passwörtern, Fingerabdruckscannern, oder Chipkarten) werden. Dies ermöglicht es, im Falle von Missbrauch entsprechende Massnahmen zu ergreifen. Darüber hinaus kann aufgrund dieser Informationen der genutzte Kurs verrechnet (z.B. zwischen verschiedenen Universitäten) werden und die Leistung des Benutzer bestätigt werden.
- **Datenschutz**
Bei einem verteilten System muss sichergestellt werden können, dass sensible Benutzerdaten, wie Passwörter oder Studienleistungen für Drittpersonen nicht ersichtlich sind. Dies kann z.B. durch sichere Netzwerkverbindungen mittels Verschlüsselung garantiert werden.
- **Abrechnung**
Dienste, die die StudentInnen in Anspruch nehmen, müssen in geeigneter Weise abgerechnet werden können um diese den StudentInnen oder der Heiminstitution der GaststudentInnen in Rechnung stellen zu können.
- **Bestätigung von Leistungen**
Nimmt eine StudentIn an einem Fernkurs teil, so hat er/sie ein Interesse daran, dass die geleisteten Arbeiten wie Übungen oder Praktika auch entsprechend bestätigt werden und in einem Leistungsausweis eingetragen werden.
- **Didaktik / Pädagogik**
Um einen hohen Lernerfolg zu erzielen, reicht es nicht, dass die vorhandenen Lehrmittel wie Skripte oder Präsentationen ohne Überarbeitung in eine Webseite übernommen werden. Da die StudentInnen bei einem Fernkurs meistens im Selbststudium die Arbeit verrichten, müssen Werkzeuge und Hilfsmittel angeboten werden, mit denen sie Hilfe bei Problemen bekommen [9][10]. Als Beispiele seien hier Foren, Chat-Räume und Videokonferenzen genannt .

2.2 Übergeordnetes Programm: Swiss Virtual Campus (SVC)

Der Swiss Virtual Campus, ein Projekt des Bundesamtes für Bildung und Wissenschaft, hat zum Ziel das Lernen via Internet auf Hochschulebene zu fördern.

Die StudentInnen sollen sich, vom Vorlesungsplan ungebunden, zeit- und ortsunabhängig Wissen aneignen können.

Die SVC Kurse bieten Lerninhalte, die via Internet verfügbar sind und ersetzen traditionelle Präsenzvorlesungen, bei denen vielfach Platzmangel in den Vorlesungssälen herrscht, durch obligatorische Fernkurse. Es ist nicht das Ziel komplette Studiengänge auf das Internet zu verlagern, sondern eine ideale Ergänzung zu den Vorlesungen und Praktika zu offerieren.

Fachleute, Pädagogik- und Didaktikexperten legen grossen Augenmerk auf die hohe Kursqualität. Die Art der Wissensvermittlung soll attraktiv und dank der Interaktivität effizienter werden", umschreibt Bernard Levrat, einer der geistigen Väter des Programms, eines der Hauptziele des SVC.

Um den besonderen Verhältnissen in der Schweiz gerecht zu werden, werden die Fernkurse – falls möglich – in mehreren Sprachen angeboten.

Aufgrund der Ausschreibungen für SVC Projekte konnten 50 Projekte [11] ausgewählt werden. Eine erste Serie von 28 Projekten wurde bereits im Herbst 2000 gestartet, während die 22 Projekte der zweiten Serie im Sommer 2001 begonnen wurden. In den 50 Projekten sind viele Fachgebiete wie Medizin, Technologie, Geisteswissenschaften, Management und Administration, Naturwissenschaften, Erziehungswissenschaften, Physik, Mathematik und Informatik sowie Wirtschafts- und Rechtswissenschaften vertreten (Aufzählung nach Anzahl Projekte sortiert). Um die Zusammenarbeit zwischen den Hochschulen zu fördern, wird jedes der 50 bewilligten Projekte von mindestens drei Hochschulen zusammen durchgeführt. Die Projektkosten werden zu mindestens 50% durch die Hochschulen (Universitäten, Eidgenössische Technische Hochschule (ETH) und Fachhochschulen) getragen. Der Rest der Kosten wird durch das Bundesamt für Bildung und Wissenschaft (für die Unis), vom ETH-Rat (für die ETH) und vom Bundesamt für Berufsbildung und Technologie (BBT, im Falle der Fachhochschulen) übernommen.

2.3 Beispiele von Internet-basierten Kursen, Kursverwaltungssystemen und Werkzeugen

2.3.1 Einfache Webkurse: W3 Schools

W3 Schools ist eine freie öffentlich-zugängliche Sammlung von Webtutorials zu HTML, XML, WML, CSS, JavaScript, ASP, SQL und Flash. Zu jedem Thema existiert ein Multiple-Choice Quiz mit 20 Fragen, welches die bisher verbrauchte Zeit anzeigt und am Schluss die Antworten des Benutzers korrigiert [13].

2.3.2 Kursverwaltungssysteme

2.3.2.1 WebCT

WebCT ist ein kostenpflichtiges, webbasiertes Kursverwaltungssystem, welches das Anbieten und Administrieren von Fernkursen erleichtert. Kurse bestehen aus Theorieteilen, Quizzes und Selbsttests. Bereits vorhandene Webseiten können in das Kurssystem integriert werden. Mittels eines Webinterfaces können Kursteilnehmer hinzugefügt, bearbeitet und überwacht werden. Alle Teilnehmer können über Diskussionsforen, Mail, Chat oder Whiteboards untereinander Information austauschen und ihre Fortschritte selbständig überprüfen. Weitere Hilfsmittel wie Syllabus, ein Stichwortverzeichnis und ein Kalender runden das System ab.

2.3.2.2 Top Class

Die TopClass E-Learning Suite [19] ist eine kommerzielle webbasierte Lernplattform, welche die Konvertierung von existierenden Lerninhalten (z.B. Präsentationen) und die Publizierung derselben ins Internet ermöglicht. Basis ist ein Dokumentenverwaltungssystem (engl. Content Management System, CMS) in welchem alle Inhalte in einem eigenen Format abgespeichert werden. Mittels Zusatzmodulen können Wissenslücken gefunden und analysiert werden oder Lerninhalte zum netzwerk-unabhängigen Studium auf portable Studentenrechner gespeichert werden. Des weitern bietet es eine automatische Kontrolle der Studentenleistungen. Leider sind über die dahinter stehenden Technologien keine Informationen verfügbar.

2.3.2.3 Lotus Learning Space

Learning Space [20], entwickelt von IBM Lotus, ist eine Sammlung von Produkten, mit dem Ziel, die Ausbildungseffizienz in Firmen zu verbessern. Das Kernmodul benützt die Active Server Pages des Microsoft Information Servers und eine relationale Datenbank um Lerninhalte zu publizieren und den Fortschritt des Lernenden zu messen. Der Lotus Learning Space Virtual Classroom ermöglicht verteiltes Lernen mittels Videokonferenzen, vorgefertigten Übungen, Chats und gemeinsamen Whiteboards.

2.3.3 Entfernte Experimentierplattformen

2.3.3.1 Mentortech

Mentor Technologies bietet ein kommerzielles, ausgereiftes Kurssystem, mit welchem Geräte, die vorgängig reserviert wurden, fernkonfiguriert werden können. Der Kurs beinhaltet Theorie, Übungen und Praktika im Bereich von Netzwerkgeräten. Aufgrund ihres Abkommens mit einem führenden Hersteller von Routern, ist es ihnen möglich, dessen ganze Produktpalette als Fernkurs anbieten zu können. Leider existieren keine öffentlich zugänglichen Details über die Technik, die hinter ihrem Kurssystem steht [14].

2.3.3.2 Emulab

In den Vereinigten Staaten von Amerika existiert ein riesiges Netzwerklabor namens Emulab [15]. Ein Teilprojekt davon ist „Emulab Classic“, in welchem die Universität von Utah Zugang zu mehreren hundert Standard-Rechnern offerieren, um darauf Netzwerkemulationen und Experimente durchzuführen. Jeder dieser Knoten enthält fünf Netzwerkkarten, die einerseits dazu dienen, die Knoten untereinander zu verbinden, und andererseits um den Zugriff auf die zentralen Server zu erlauben.

Im Mittelpunkt des ganzen Systems steht ein programmierbares Patchpanel, welches es ermöglicht, nahezu beliebige Netzwerktopologien zu realisieren. Im Rahmen eines Experiments können registrierte StudentInnen mittels des Netzwerksimulators (ns-2) [16] Topologien, bestehend aus Routern, Traffic-Generatoren und Endgeräten, erstellen die dann auf die echten Knoten abgebildet werden.

Während eines solchen Experiments erhalten die StudentInnen uneingeschränkten (inklusive Administratorrechten) exklusiven Zugriff auf die Knoten. Der Hauptunterschied zu unserem System besteht darin, dass keine realen Netzwerkkomponenten wie zum Beispiel Router angeboten werden, sondern dass diese auf den Standard-Rechnern emuliert werden. Es versteht sich von selbst, dass sich Experimente mittels echten Hardwarekomponenten näher an der Realität befinden, als Emulationen.

2.3.3.3 Nano-World

Ein weiteres Swiss Virtual Campus Projekt nennt sich Nano-World [17] und vermittelt einen Einblick in Nanotechnologien. Nano-World bietet Fernzugriff auf Elektronenmikroskope. Ihre Anforderungen sind denen von VITELS nahe verwandt und so ist auch ihr Lösungsansatz.

2.3.4 Werkzeuge für verteiltes Lehren: ARIADNE

Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) [18] ist ein europäisches Forschungs- und Technologieentwicklungsprojekt. Es auf die Entwicklung von Werkzeugen und Methoden ausgerichtet, die das Erstellen, Bearbeiten und Wiederverwenden von Rechner-basierten pädagogischen Inhalten ermöglicht. Informationen werden in sogenannten Wissenspools gespeichert und es bietet individuelle Kursansichten, das heisst das Erscheinungsbild des Kurses passt sich den StudentInnenleistungen an. Das Projekt hat zum Ziel, StudentInnen und Dozenten effiziente und motivierende Lernszenarios zur Verfügung zu stellen und die Kommunikation und die Zusammenarbeit zwischen Forschern zu verbessern. Hierzu wurden Werkzeuge wie das Pedagogic Header Generator & Validation Tool oder das Query Tool entwickelt. Das erstere ermöglicht es, Inhalte mit Zusatzinformationen, wie potentielle Zuhörer, Schwierigkeit, oder Dauer des Lerninhaltes zu versehen und diese zu validieren. Mit dem zweiten Werkzeug können Dokumente aufgrund eben dieser Zusatzinformationen gesucht und ausgewählt werden.

3 Grundlegende Techniken

3.1 Betriebssysteme und Server

3.1.1 Linux

Linux [21] ist ein UNIX [23]-ähnliches Betriebssystem, welches ursprünglich von Linus Torvalds entwickelt wurde. Linux ist Open-Source [22] und Freeware, also gratis, kann aber auch verkauft werden, um die Kosten für gedruckte Handbücher, Medien (CDs und/oder DVDs) und Support zu decken. Zuerst für x86-basierte Rechner entwickelt, kann es heute auf vielen verschiedenen Plattformen betrieben werden. Wegen seiner Stabilität, der sehr guten Netzwerkunterstützung und der sparsamen Nutzung von Rechnerressourcen ist es für Serverdienste wie Webserver sehr beliebt. Die in dieser Arbeit beschriebenen Rechner werden alle mit Linux betrieben

3.1.2 Apache

Apache [24] ist ein Webserver, der auf der Basis des NCSA HTTP Servers [25] weiterentwickelt wurde. Apache ist der meistverbreitete Webserver, ist Open-Source und Freeware. Er ist für Linux, Windows und mehrere Versionen von UNIX erhältlich.

3.1.3 Lightweight Directory Access Protocol (LDAP)

Das Lightweight Directory Access Protocol (LDAP) ist ein Client / Server-Protokoll, mittels welchem auf Verzeichnisdienste zugegriffen werden kann. Ursprünglich als Frontend des X.500 Protokolls [33] benutzt, kann es auch mit eigenständigen Verzeichnisdiensten benutzt werden. LDAP ist für Lesenden Zugriff optimiert und wird daher meistens in Umgebungen gebraucht, wo sich Daten selten ändern. Das Projekt OpenLDAP [34] bietet eine Open-Source-Version von LDAP an.

3.2 Programmiersprachen

3.2.1 PHP

Der Hypertext Preprocessor (PHP) [26] ist eine server-seitige Skriptsprache, welche innerhalb von HTML-Seiten eingebettet werden kann und – falls der Webserver diese unterstützt – dazu benutzt werden kann, um dynamische Webinhalte zu generieren. PHP bietet Schnittstellen zu Datenbanken wie MySQL [27], PostgreSQL [28] und Oracle [29] sowie die Möglichkeit auf LDAP-Server [32] oder das darunterliegende Betriebssystem zuzugreifen. Das Webfrontend des IPSec-Moduls wurde in PHP geschrieben.

3.2.2 Perl

Die Skriptsprache PERL (Practical Extraction and Report Language) [30] wurde Mitte der achtziger Jahren vom Systemadministrator Larry Wall entwickelt, um damit Administrationsaufgaben einfacher zu gestalten. Perl zeichnet sich durch eine hohe Funktionenvielfalt für das Suchen und Ersetzen von Text-Mustern aus und wurde und wird vielfach für die Programmierung von Skripten auf Webservern verwendet, die das Common Gateway Interface (CGI) [31] nutzen. Zu Perl gibt es Tausende von Zusatzbibliotheken für die verschiedensten Anwendungen. In Perl wurden Skripte programmiert, die auf dem Portal ausgeführt werden und auf die Router zugreifen.

3.3 Sicherheitsrelevante Protokolle

3.3.1 Secure sockets Layer (SSL) / Transport Layer Security (TLS)

Das Secure sockets Layer Protokoll (SSL) [35] wurde von Netscape entwickelt und dient dazu, eine verschlüsselte sichere End-zu-End-Verbindung zwischen zwei Kommunikationspartnern zu ermöglichen. Darüber hinaus unterstützt es die Authentifizierung des Diensteanbieters (Server) und optional die des Klientrechners (Client). SSL wird auf einem verlässlichen Transport Protokoll wie z.B. TCP [36] aufgesetzt. Der Hauptvorteil von SSL ist der, dass es Protokollunabhängig ist und damit einem übergeordneten Applikationsprotokoll wie z.B. HTTP [37], FTP [38], Telnet [39] völlige Transparenz bietet. Verbindungen via SSL sind ausserdem verlässlich, denn eine Veränderung eines Netzwerkpaketes kann durch Prüfsummen in Form von sicheren Einweg-Funktionen detektiert werden.

Transport Layer Security (TLS) [40] ist die neuste verfügbare Erweiterung von SSL und basiert auf der von Netscape spezifizierten Version 3.0 von SSL. Die Unterschiede zwischen TLS 1.0 und SSL 3.0 sind insofern markant, als die beiden Protokolle zueinander nicht kompatibel sind. Des weitem unterstützt TLS stärkere Verschlüsselung als SSL. Das Projekt OpenSSL [41] bietet eine Open-Source-Implementierung der SSL / TLS-Protokolle an, welche in dieser Arbeit benützt wird. Ein Beispiel für ein auf SSL / TLS aufbauendes Protokoll ist das im folgenden beschriebene Hyper Text Transport Protocol over Secure Sockets Layer.

Hyper Text Transport Protocol over Secure Sockets Layer (HTTPS) [42] bietet kryptographisch-sichere Verbindungen zwischen Webservern und Browsern und ermöglicht es dadurch private Daten (z.B. Kontonummern, Passwörter, Kreditkartennummern) auszutauschen, ohne das Risiko des Mitlesens durch dritte einzugehen. Dies ist die Grundlage um Bankgeschäfte übers Internet zu erledigen oder via Kreditkarte bei einem Internetgeschäft einzukaufen. Die heutzutage vorhandenen Browser unterstützen alle dieses Protokoll ohne zusätzliche Installationen. Für die Anbieter von sicheren Webinhalten existieren SSL/TLS-Module für alle geläufigen Server, wie z.B. den Apache oder den Microsoft Internet Information Server (IIS) [43]. In dieser Arbeit wird ein Apache Server mit dem Modul `mod_ssl` [44] verwendet, um die Übertragung zwischen Student und Universität zu sichern. Das Projekt `mod_ssl` baut auf SSL / TLS auf und basiert auf der obenerwähnten OpenSSL-Bibliothek.

3.3.2 Secure Shell (SSH)

Um über ein Netzwerk, z.B. das Internet auf einem entfernten Rechner zu arbeiten, wurde bis anhin Telnet verwendet. Telnet hat jedoch den grossen Nachteil, dass Benutzernamen und Passwörter im Klartext, das heisst unverschlüsselt übertragen werden. Dadurch können diese Daten durch Dritte, die Zugriff auf das Netzwerk haben, mitgelesen werden. Daher wird heute meistens die Secure Shell (SSH) verwendet, welche die Telnet-Funktionalität über eine gesicherte Verbindung

ermöglicht. Secure Shell Server und Client werden als kommerzielles Produkt für Windows- und Unix- bzw. Linux-basierte Betriebssysteme von SSH Communication Security [45] angeboten, es existiert jedoch mit dem Projekt OpenSSH [46] auch eine freie Implementierung des SSH-Protokolls. Die meisten Linux-Distributionen enthalten einen SSH-Dienst und einen SSH-Client. Das im folgenden beschriebene Produkt Mindterm ist ein solcher SSH-Client.

Mindterm [47] ist ein rein Java-basierter [48] SSH-Client, welcher von der Firma Appgate als Applet [49] angeboten wird. Mindterm kann als eigenständige (standalone) Applikation gestartet werden oder als Applet innerhalb einer Webseite. Als Besonderheit bietet Appgate ein sogenanntes signiertes Applet an, welches nicht nur Verbindungen zum Server herstellen darf, von dem es heruntergeladen wurde, sondern zusätzlich SSH-Verbindungen zu beliebigen Rechnern initiieren kann, was bei unsignierten Applets von der Sandbox [50] in der Java Virtual Machine [51] unterbunden wird. Mindterm ist für nicht-kommerzielle Institutionen wie Universitäten und private Personen kostenlos erhältlich. In dieser Arbeit wird die Mindterm-Applikation benutzt, um vom Studentenrechner eine sichere Verbindung auf das Portal herzustellen.

3.3.3 Virtuelle Private Netzwerke (VPN)

Ein Virtuelles Privates Netzwerk (VPN) [55] ist ein privates Datennetzwerk, welches eine öffentliche Telekommunikationsstruktur wie z.B. das Internet benützt. Der Schutz der privaten Daten wird durch ein Tunnel-Protokoll und Sicherheits-Mechanismen wie z.B. IPSec erreicht. Ein VPN ist mit einem System von eigenen oder Mietleitungen vergleichbar, auf welches nur eine Firma oder Institution Zugriff hat. Gegenüber Mietleitungen bieten VPNs den Vorteil, dass sie wesentlich günstiger sind, da sie auf bestehenden öffentlichen Infrastrukturen basieren.

Ein VPN besteht im Allgemeinen aus den folgenden Komponenten:

- **Einem internen Netzwerk 1**
Dieses befindet sich an einem beliebigen Standort A.
- **Einem VPN-Gateway 1**
Das Gateway verbindet das interne Netzwerk 1 am Standort A mit dem öffentlichen Netzwerk. Es nimmt Daten vom internen Netzwerk entgegen, verschlüsselt diese und sendet sie an eine genau definierte Gegenstelle (Gateway 2) über das öffentliche Netzwerk.
- **Einem internen Netzwerk 2**
Dieses befindet sich an einem beliebigen Standort B.
- **Einem VPN-Gateway 2**
Dieses verbindet das öffentliche Netzwerk mit dem internen Netzwerk 2 am Standort B. Daten, die vom bekannten Gateway 1 herkommen, werden akzeptiert, entschlüsselt und an das interne weitergereicht.
- **Einem öffentlichen Netzwerk**
z.B. das Internet

Virtuelle Private Netzwerke sind die Basis um Netzwerke von Firmenniederlassungen kostengünstig und sicher miteinander zu verbinden und sogenannte Wide-Area-Intranets zu bilden. Der Zugriff eines Rechners des Netzwerks 1 auf Daten des Netzwerks 2 und umgekehrt ist für diese völlig transparent. Die hinter den Gateways stehenden Rechner müssen nicht speziell konfiguriert werden. Obwohl die Daten die zwischen den Gateways ausgetauscht werden, innerhalb des öffentlichen Netzwerks unter Umständen mitgelesen werden können, ist dies kein Sicherheitsproblem, da ja nur die bereits verschlüsselten Inhalte sichtbar sind. Aufgrund der in den Paketen integrierten Prüfsummen ist es ausserdem praktisch unmöglich, Pakete abzufangen, zu verändern und wieder einzuspeisen, da dies detektiert würde.

Die folgende Abbildung 1 zeigt ein solches Szenario, bei welchem auf der Basis von IPSec-Gateways und unter Benutzung des Internets, zwei interne Netzwerke zu einem VPN verbunden sind.

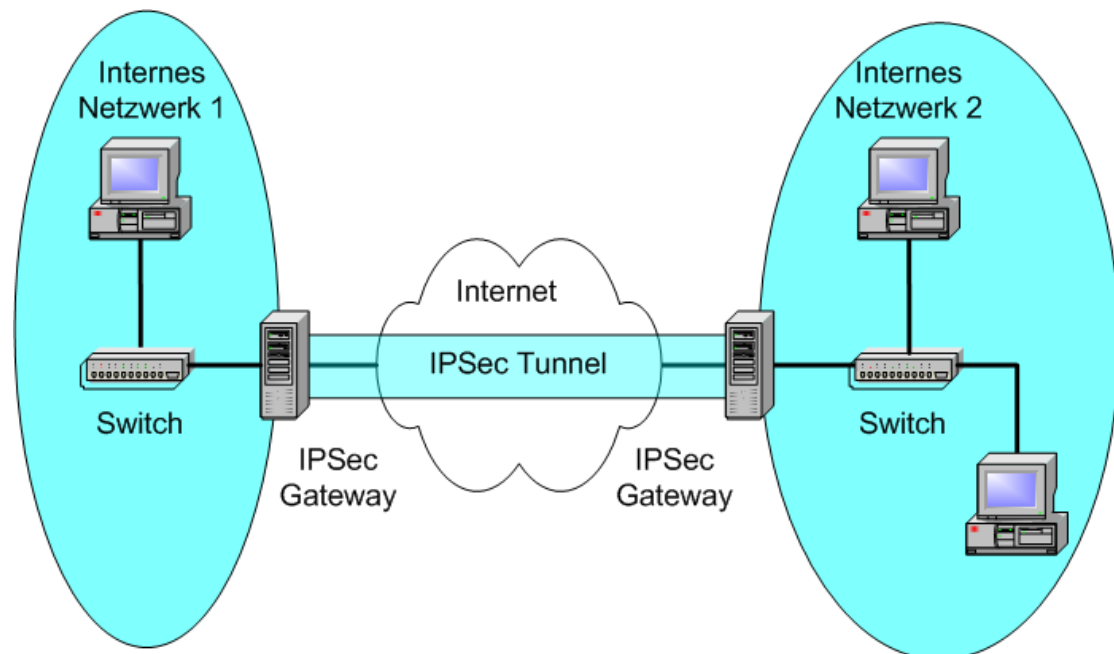


Abbildung 1: Virtuelles Privates Netzwerk

IPSec steht für Internet Protocol Security und wurde von der Internet Engineering Task Force (IETF) [52] entwickelt. IPSec wird ein Teil des Internet Protokolls Version 6 (IPv6) [53] sein. IPSec benützt starke Kryptographie um auf Protokollebene Authentifizierung und Datenintegrität anzubieten. Mittels IPSec ist es möglich, einen sicheren privaten Netzwerkkanal über ein unsicheres öffentliches Netzwerk zu etablieren. Dies wird erreicht, indem jedes Paket, von einem IPSec Gateway verschlüsselt wird, bevor es über das unsichere Medium übertragen wird. Auf der Seite der Gegenstelle befindet sich ein weiterer IPSec Gateway, welcher die verschlüsselten Pakete entgegen nimmt, diese entschlüsselt und an das sich dahinter befindende Netzwerk weiterreicht. Die im Kapitel 5 beschriebene Architektur benützt die freie IPSec Implementierung FreeS/WAN [54] für Linux um Verbindungen zwischen verschiedenen Rechnern zu sichern.

4 Das traditionelle Modul „IPSec“ des Computernetze-Praktikums

Das im Rahmen dieser Arbeit exemplarisch implementierte Fernkursmodul „IPSec“ hat seinen Ursprung im traditionellen Computernetze-Praktikum, welches jährlich als Spezialvorlesung für Hauptfachinformatiker durchgeführt wird. In diesem über ein ganzes Semester dauernden Praktikum lernen die StudentInnen verschiedene Teilaspekte der Netzwerkinstallation und Konfiguration kennen. Hierzu werden ihnen im Labor sechs Sun-Rechner, zwei Cisco Router sowie Repeater, Switches und Netzwerkkabel zur Verfügung gestellt. Zusätzlich existiert ein Laborserver, der für jedes Praktikumsmodul eine entsprechende Liste von Softwarepaketen besitzt (z.B. den Apache Webserver für das Modul 5 oder den Mailserver Sendmail für das Modul 6). Wird nun ein Arbeitsrechner mit dem Laborserver verbunden und eingeschaltet, nimmt er Kontakt zum Laborserver auf und lädt die modulspezifischen Softwarepakete herunter und installiert diese. Dies hat den Vorteil, dass man den StudentInnen problemlos volle Administrationsrechte auf den Rechnern geben kann, da bei Fehlkonfiguration der Rechner einfach nur neu gestartet werden muss.

Des Weiteren ist das Praktikumsnetzwerk vollständig vom Universitätsnetzwerk abgetrennt.

Die StudentInnen reservieren sich den Praktikumsraum (das Labor) in Gruppen von zwei bis drei Personen und bearbeiten sieben Praktikumsmodule. Dabei lernen sie wie man

1. Netzwerke mit Repeater und Switches aufbaut und testet
2. Cisco Router konfiguriert
3. Virtual Private Networks aufbaut und Netzwerke mittels SNMP verwaltet
4. einen Domain Name Service konfiguriert und betreibt
5. einen Webserver und Proxyserver konfiguriert
6. einen Mailserver konfiguriert und betreibt
7. Remote Procedure Calls programmiert

Da ein Teil dieser Diplomarbeit darin bestand, das traditionelle Praktikumsmodul 3 „IPSec“, bei dem mittels Cisco Router ein sogenanntes Virtuelles Privates Netzwerk aufgebaut wird, für den Fernzugriff anzupassen und als Fernkursmodul im Rahmen von VITELS anzubieten, wird dieses im folgenden erklärt.

4.1 Ziel

Das Praktikumsmodul „IPSec“ hat zum Ziel, den StudentInnen den Umgang und die Konfiguration von Cisco Router näher zu bringen und ihnen zu ermöglichen, damit ein Virtuelles Privates Netzwerk aufzubauen. Mit Werkzeugen wie Netpipe (ein Netzwerkbandbreitenmessprogramm) und Tcpcdump (ein Netzwerk-„Sniffer“). Dies ist eine Applikation, die es ermöglicht, Pakete mitzulesen, die an der Netzwerkschnittstelle des Rechners vorbeikommen, auf dem Tcpcdump läuft), vergleichen sie die Netzwerkeigenschaften mit und ohne etabliertem VPN.

4.2 Ablauf

Der Ablauf des Praktikummoduls „IPSec“ gestaltet sich wie folgt:

- Aufbau des Netzwerks
- Konfiguration der Linux-Rechner
- Konfiguration der Router
- Testen des Netzwerks und Messen der Bandbreitenkapazität, Mitlesen des Netzwerkverkehrs zwischen Netzwerk 1 und Netzwerk 2 auf dem Rechner host2
- Konfiguration des VPN-Tunnels zwischen den Routern
- Testen des Netzwerks und Messen der Bandbreitenkapazität, Mitlesen des Netzwerkverkehrs zwischen Netzwerk 1 und Netzwerk 2 auf dem Rechner host2

4.3 Aufbau

Für das „IPSec“ Modul stehen die folgenden Hardwarekomponenten zur Verfügung:

- Ein Cisco 2620 Router
- Ein Cisco 3620 Router
- Drei Sun-Maschinen mit Debian Linux
- Drei Repeater
- Netzwerkkabel

In einen ersten Schritt müssen die StudentInnen die zur Verfügung stehenden Hardwarekomponenten gemäss der Abbildung 2 aufbauen und verbinden.

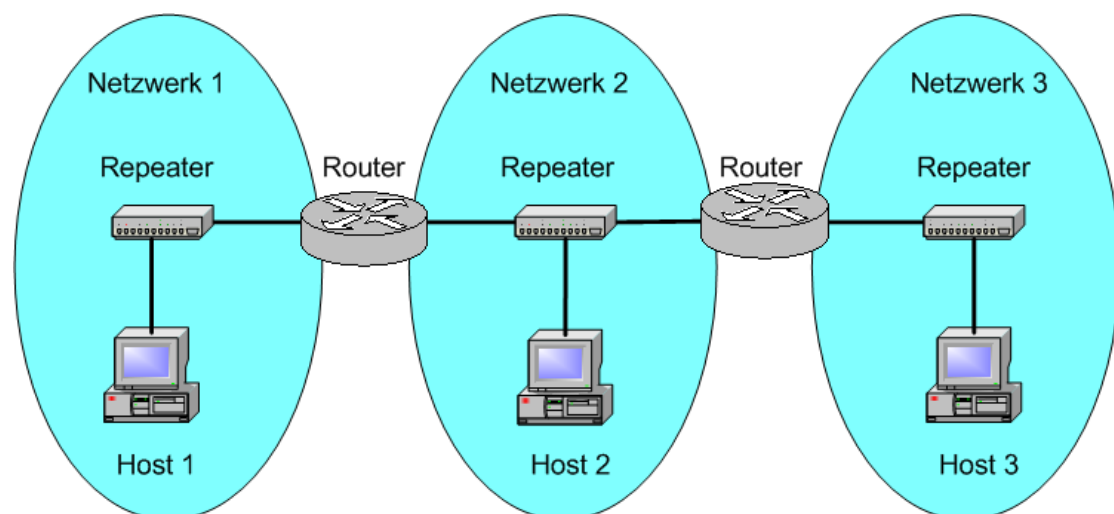


Abbildung 2: Aufbau des traditionellen IPSec Moduls

4.4 Konfiguration der Linux-Rechner

Um die Linux-Rechner konfigurieren zu können, müssen sich die StudentInnen lokal am Rechner als Administrator (root) anmelden. Das Administrationspasswort wurde vorgängig bekannt gegeben und wird durch die Konfiguration auf dem Laborserver festgelegt.

Die Netzwerkkonfiguration der Linux-Rechner wird mittels der beiden Kommandozeilenbefehle **ifconfig** und **route** durchgeführt.

4.4.1 Netzwerkkarten-Konfiguration mit ifconfig

Mit dem Befehl **ifconfig**, dessen Ausführung Administratorrechte benötigt, werden Netzwerkschnittstellen wie Netzwerkkarten konfiguriert oder deren momentane Konfiguration angezeigt. Die Syntax ist wie folgt:

```
ifconfig <interface> <a.b.c.d> netmask <e.f.g.h>  
broadcast <i.j.k.l> up
```

wobei <interface> die Netzwerkschnittstelle, <a.b.c.d> die IP-Adresse, <e.f.g.h> die Netzwerkmaske und <i.j.k.l> die Broadcast-Adresse definiert. Durch up wird die Schnittstelle aktiviert.

Die Linux-Rechner müssen danach gemäss der Tabelle 2 konfiguriert werden, sodass sich jeder Rechner in einem eigenen Subnetzwerk befindet.

Host 1	Router 1		Host 2	Router 1		Host 3
Netzwerk-Karte 1	Netzwerk-Karte 1	Netzwerk-Karte 2	Netzwerk-Karte 1	Netzwerk-Karte 1	Netzwerk-Karte 2	Netzwerk-Karte 1
10.1.0.100	10.1.0.10	10.2.0.10	10.2.0.100	10.2.0.20	10.3.0.20	10.3.0.100

Tabelle 2: Zuweisung der IP-Adressen

4.4.2 IP-Routing-Konfiguration mit route

Damit die Rechner Subnetz-übergreifend Pakete verschicken und empfangen können müssen IP-Routen definiert werden. Dies wird mittels des Befehls **route** konfiguriert, dessen Syntax wie folgt aussieht:

```
route add -net <target> gw <gateway> dev <interface>
```

wobei add angibt, dass eine Route hinzugefügt wird, <target> das Zielnetz dieser Route definiert, <gateway> den Gateway definiert, der das Netz in dem sich der Rechner befindet mit dem Zielnetz verbindet und <interface> angibt über welche Schnittstelle die Pakete versendet werden sollen.

Ein spezielles Ziel ist `default`. Diese Route kommt dann zum Zuge, falls keine andere explizit-definierte Route auf ein gegebenes Paket passt.

Die Routen der Linux-Rechner werden gemäss der Tabelle 3 konfiguriert.

	Destination	Gateway
Host 1	10.1.0.0	*
	Default	10.1.0.10
Host 2	10.2.0.0	*
Host 3	10.3.0.0	*
	Default	10.3.0.20

Tabelle 3: Routingtabelle der Linux-Rechner

4.5 Konfiguration der Cisco Router

4.5.1 Zugriff über Netzwerk / Konsole

Die beiden vorhandenen Cisco Router besitzen je zwei Ethernet-Netzwerkschnittstellen und einen seriellen Anschluss, den Con-Port. Die Netzwerkschnittstellen können ähnlich wie unter Linux mittels Kommandozeilen-Befehle konfiguriert werden. Falls mindestens eine Netzwerkschnittstelle konfiguriert ist, kann man sich von einem beliebigen Rechner, der den Befehl Telnet (vergleiche Kapitel 3.3.2) unterstützt, am Router anmelden und ihn konfigurieren.

Sind die Schnittstellen jedoch nicht oder fehlerhaft konfiguriert, muss die serielle Schnittstelle eines Rechners (beispielsweise eines portablen Rechners) mit dem Konsolenport (Con-Port) des Routers verbunden werden und auf dem Rechner eine Terminalapplikation wie beispielsweise Minicom gestartet werden.

4.5.2 Minicom

Die Terminalapplikation Minicom wurde früher dazu benutzt, um via ein an der seriellen Schnittstelle angeschlossenes Modem eine Verbindung zu einem entfernten Rechner über das normale Telefonnetzwerk aufzubauen. Seit der Existenz des Internets wird es meistens nur noch zu Kontrollzwecken von an seriellen Schnittstellen angeschlossenen Geräten, wie zum Beispiel Modems gebraucht, oder eben um damit einen Router über den Konsolenport anzusprechen und zu konfigurieren.

4.5.3 Netzwerk-Konfiguration der Router

Nachdem sich der Student mittels der Terminalapplikation mit dem Router verbunden hat, kann er diesen konfigurieren. Um den Netzwerk-Schnittstellen die IP-Adressen der Tabelle 2 zu zuweisen, muss in den normalerweise passwort-geschützten EXEC-Modus gewechselt werden, was durch Eingabe des Kommandos `enable` bewirkt wird. Durch Eingabe von `configure` wechselt man in den Konfigurationsmodus. Mit den Befehlen `interface`, `ip address`, `ip broadcast-address` und `no shutdown` werden die IP-Adressen den Schnittstellen zugewiesen. Die folgende Abbildung 3 zeigt den dazugehörigen Dialog mit dem Router 2600:

```
Router>
Router>enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#hostname cisco2600
cisco2600(config)#interface FastEthernet 0/0
cisco2600(config-if)#ip address 10.2.0.10 255.255.255.0
cisco2600(config-if)#ip broadcast-address 10.2.0.255
cisco2600(config-if)#no shutdown
cisco2600(config-if)#exit
cisco2600(config)#interface FastEthernet 0/1
cisco2600(config-if)#ip address 10.1.0.10 255.255.255.0
cisco2600(config-if)#ip broadcast-address 10.1.0.255
cisco2600(config-if)#no shutdown
cisco2600(config-if)#exit
cisco2600(config)#
```

Abbildung 3: IP-Adress-Konfiguration des Routers 2600

Die Konfiguration der IP-Routen der Cisco Router gestaltet sich relativ simpel, da diese das Routing Information Protocol (RIP) unterstützen, mit welchem die Router in der Lage sind, ihre Routing-Tabellen selbständig aneinander anzupassen. Dies wird durch die Befehle in der Abbildung 4 erreicht:

```
cisco2600(config)#ip routing
cisco2600(config)#router rip
cisco2600(config-router)#network 10.1.0.0
cisco2600(config-router)#network 10.2.0.0
cisco2600(config-router)#neighbor 10.2.0.20
cisco2600(config-router)#version 2
cisco2600(config-router)#exit
cisco2600(config)#
cisco2600(config)#interface FastEthernet 0/0
```

```
cisco2600(config-if)#ip rip send version 2
cisco2600(config-if)#ip rip receive version 2
cisco2600(config-if)#
cisco2600(config-if)#
```

Abbildung 4: RIP-Konfiguration des Routers 2600

Danach wird der zweite Router in gleicher Weise wie der erste konfiguriert. Dies wird durch die folgenden beiden Abbildungen (Abbildung 5 und Abbildung 6) illustriert.

```
Router>enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
terminal
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)#hostname cisco3600
cisco3600(config)#interface Ethernet 0/0
cisco3600(config-if)#ip address 10.2.0.20 255.255.255.0
cisco3600(config-if)#ip broadcast-address 10.2.0.255
cisco3600(config-if)#no shutdown
cisco3600(config-if)#exit
cisco3600(config)#exit
cisco3600#configure
Configuring from terminal, memory, or network [terminal]?
terminal
Enter configuration commands, one per line. End with
CNTL/Z.
cisco3600(config)#hostname cisco3600
cisco3600(config)#interface Ethernet 1/0
cisco3600(config-if)#ip address 10.3.0.20 255.255.255.0
cisco3600(config-if)#ip broadcast-address 10.3.0.255
cisco3600(config-if)#no shutdown
cisco3600(config-if)#exit
cisco3600(config)#
```

Abbildung 5: IP-Adress-Konfiguration des Routers 3600

```
cisco3600(config)#ip routing
cisco3600(config)#router rip
cisco3600(config-router)#network 10.2.0.0
cisco3600(config-router)#network 10.3.0.0
cisco3600(config-router)#neighbor 10.2.0.10
cisco3600(config-router)#version 2
cisco3600(config-router)#exit
cisco3600(config)#
cisco3600(config)#interface Ethernet 0/0
cisco3600(config-if)#ip rip send version 2
cisco3600(config-if)#ip rip receive version 2
cisco3600(config-if)#
```

Abbildung 6: RIP-Konfiguration des Routers 3600

4.6 Testen der Konfiguration mit Ping und Traceroute

Nachdem der zweite Router entsprechend des vorigen Kapitels 4.5.3 konfiguriert wurde, kann das erstellte Netzwerk-Szenario mit den Werkzeugen Ping [59] und Traceroute [60] getestet werden. Ping sendet einen Internet Control Message Protocol (ICMP)-Request [61] an einen anzugebenden entfernten Rechner und wartet darauf, dass dieser mit einem ICMP-Reply antwortet. Ping berechnet das Verhältnis zwischen gesendeten und empfangenen Paketen und misst die Zeit, die das Paket benötigt, um vom Sender zum Empfänger und zurück zu wandern. Traceroute zeigt alle Stationen an, die ein Paket vom sendenden Rechner zum Empfänger passiert. Die folgenden Abbildungen zeigen die dafür notwendigen Befehle:

```
host1@host1:~$ ping host3
PING host3 (10.3.0.100): 56 data bytes
64 bytes from 10.3.0.100: icmp_seq=0 ttl=253 time=1.2 ms
64 bytes from 10.3.0.100: icmp_seq=1 ttl=253 time=1.1 ms
64 bytes from 10.3.0.100: icmp_seq=2 ttl=253 time=1.1 ms

--- host3 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.2 ms
host1@host1:~$
```

Abbildung 7: Pingen des host3 vom host1 aus

```
host1@host1:~$ traceroute host3
traceroute to host3 (10.3.0.100), 30 hops max, 38 byte
packets
 1  cisco2600_1 (10.1.0.10)  0.982 ms  0.998 ms  0.915 ms
 2  cisco3600_0 (10.2.0.20)  2.040 ms  1.722 ms  1.683 ms
 3  host3 (10.3.0.100)  1.021 ms  0.864 ms  0.822 ms
host1@host1:~$
```

Abbildung 8: Traceroute vom host1 zum host3

```
host3@host3:~$ ping host1
PING host1 (10.1.0.100): 56 data bytes
64 bytes from 10.1.0.100: icmp_seq=0 ttl=253 time=1.2 ms
64 bytes from 10.1.0.100: icmp_seq=1 ttl=253 time=1.1 ms
64 bytes from 10.1.0.100: icmp_seq=2 ttl=253 time=1.1 ms

--- host1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.2 ms
host3@host3:~$
```

Abbildung 9: Pingen des host1 vom host3 aus

```
host3@host3:~$ traceroute host1
traceroute to host1 (10.1.0.100), 30 hops max, 38 byte
packets
 1  cisco3600_1 (10.3.0.20)  1.541 ms  1.376 ms  1.361 ms
 2  cisco2600_0 (10.2.0.10)  1.361 ms  1.156 ms  1.086 ms
 3  host1 (10.1.0.100)  0.862 ms  0.865 ms  0.826 ms
host3@host3:~$
```

Abbildung 10: Traceroute vom host3 zum host1

4.7 Bandbreitenmessung mit Netpipe

Netpipe [62] ist eine freie Applikation, welche auf zwei Rechnern, die durch ein Netzwerk miteinander verbunden sind, gestartet wird, um damit die Kapazität der Netzwerkbandbreite zu messen. Während die beiden Prozesse laufen, wird sukzessive die Grösse der übertragenen Pakete in einem wählbaren Intervall erhöht. Zu dem können eine minimale Anfangs- und eine maximale Endpaketgrösse angegeben werden. Durch diese Parametrisierbarkeit ist es elegant möglich, sich an die maximale Bandbreite des zu testenden Netzwerks heranzutasten. Im traditionellen Praktikumsmodul IPsec wird dieses Werkzeug dazu verwendet, um die Unterschiede der Bandbreite zwischen den Routern mit und ohne VPN-Tunnel festzuhalten.

4.8 Netzwerk-Sniffing mit Tcpdump

Tcpdump [63] ist ein Werkzeug, welches es ermöglicht, die Pakete die an der Netzwerkschnittstelle des Rechners vorbeikommen, auf welchem die Applikation gestartet wird, mitzulesen (engl Sniffing). Tcpdump bietet eine Vielfalt von Optionen um die angezeigten Pakete auf bestimmte Ports, Quell- und Zielrechner und Protokolle einzuschränken. Im traditionellen IPsec wird es auf dem Rechner host2 dazu verwendet, um zu zeigen, dass ohne den VPN-Tunnel der Verkehr zwischen den Rechnern host1 und host3 im Klartext übertragen wird. Nach der Konfiguration des VPN-Tunnels überzeugen sich die StudentInnen in einem zweiten Schritt davon, dass der Verkehr nun durch den Tunnel geschützt ist und z.B. Passwörter die mittels Telnet ungeschützt übertragen werden, nicht mehr mitgelesen werden können.

4.9 Konfiguration des VPN-Tunnels zwischen den Routern

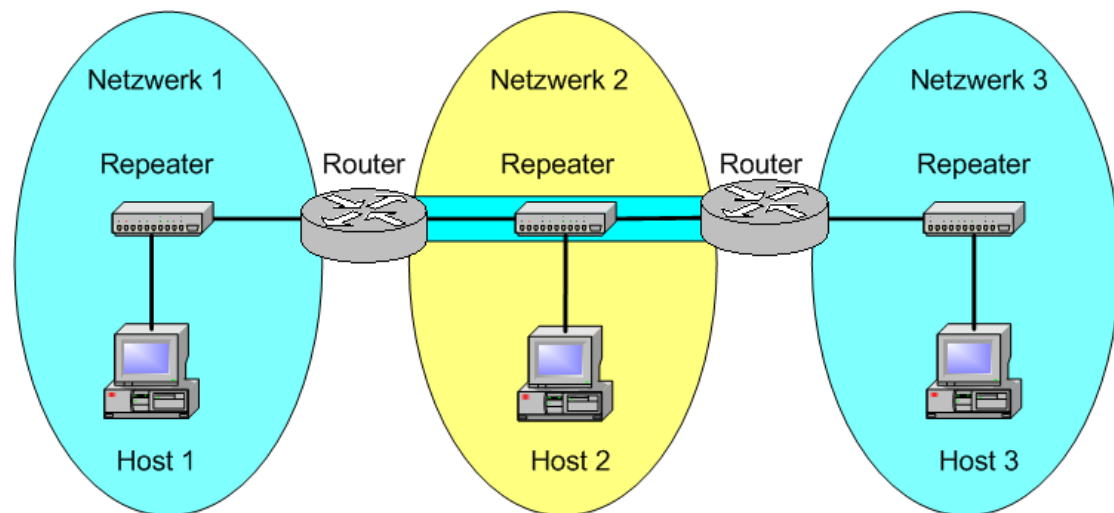


Abbildung 11: Verbindung zweier Netze mittels VPN-Tunnel

Um mit den beiden Cisco Routern einen VPN-Tunnel aufzubauen, müssen die folgenden Schritte vollzogen werden:

- **Generieren der Schlüssel**

Dies wird gemäss der Abbildung 12 durchgeführt.

- **Austauschen der Schlüssel**

Beim Schlüsselaustausch wird der eine Router in den passiven Modus geschaltet und wartet darauf, dass er einen Schlüssel empfängt. Dem zweiten Router wird die IP-Adresse des ersten Routers angegeben. Anschliessend tauschen die Router ihre Schlüssel aus und speichern diese nach Bestätigung ab (Abbildung 13).

- **Konfigurieren der Router**

Der Tunnel wird mittels sogenannter Accesslists (Zugriffslisten) aufgebaut. Jedem Router muss sein Gegenüber („peer“) und das dahinter liegende Netzwerk angegeben werden. Die notwendigen Schritte sind in der Abbildung 14 festgehalten.

```
cisco2600>enable
cisco2600#configure
Configuring from terminal, memory, or network [terminal]?
terminal
Enter configuration commands, one per line. End with
CNTL/Z.
```

```

cisco2600(config)#crypto key generate dss cisco2600
Generating DSS keys ....
[OK]

cisco3600>enable
cisco3600#configure
Configuring from terminal, memory, or network [terminal]?
terminal
Enter configuration commands, one per line. End with
CNTL/Z.
cisco3600(config)#crypto key generate dss cisco3600
Generating DSS keys ....
[OK]

```

Abbildung 12: Generieren der Schlüssel

```

cisco2600(config)#crypto key exchange dss passive
Enter escape character to abort if connection does not
complete.
Wait for connection from peer[confirm]
Waiting ....
Public key for cisco3600:
  Serial Number 012AFB17
  Fingerprint   76CF 2860 F67B 0BA7 174E

Add this public key to the configuration? [yes/no]: yes
Send peer a key in return[confirm]
Which one?

cisco2600? [yes]:
Public key for cisco2600:
  Serial Number B39FD8CE
  Fingerprint   C486 5976 FF23 2F94 E12D

cisco2600(config)#

cisco3600(config)#crypto key exchange dss 10.2.0.10
cisco3600
Public key for cisco3600:
  Serial Number 012AFB17
  Fingerprint   76CF 2860 F67B 0BA7 174E

Wait for peer to send a key[confirm]
Waiting ....
Public key for cisco2600:
  Serial Number B39FD8CE
  Fingerprint   C486 5976 FF23 2F94 E12D

Add this public key to the configuration? [yes/no]: yes
cisco3600(config)#

```

```
cisco3600 (config) #
```

Abbildung 13: Austauschen der Schlüssel

```
cisco2600 (config) #ip access-list extended 100
cisco2600 (config-ext-nacl) #permit tcp 10.1.0.0 0.0.0.255
10.3.0.0 0.0.0.255
cisco2600 (config-ext-nacl) #permit udp 10.1.0.0 0.0.0.255
10.3.0.0 0.0.0.255
cisco2600 (config-ext-nacl) #exit
cisco2600 (config) #crypto map cisco3600 200
% NOTE: This new crypto map will remain disabled until a
peer
and a valid access list have been configured.
cisco2600 (config-crypto-map) #set peer cisco3600
cisco2600 (config-crypto-map) #match address 100
cisco2600 (config-crypto-map) #set algorithm des cfb-64
cisco2600 (config-crypto-map) #exit
cisco2600 (config) #interface FastEthernet 0/0
cisco2600 (config-if) #crypto map cisco3600
cisco2600 (config-if) #exit
cisco2600 (config) #exit
cisco2600 #

cisco3600 (config) #ip access-list extended 100
cisco3600 (config-ext-nacl) #permit tcp 10.3.0.0 0.0.0.255
10.1.0.0 0.0.0.255
cisco3600 (config-ext-nacl) #permit udp 10.3.0.0 0.0.0.255
10.1.0.0 0.0.0.255
cisco3600 (config-ext-nacl) #exit
cisco3600 (config) #crypto map cisco2600 300
cisco3600 (config-crypto-map) #set peer cisco2600
cisco3600 (config-crypto-map) #match address 100
cisco3600 (config-crypto-map) #set algorithm des cfb-64
cisco3600 (config-crypto-map) #exit
cisco3600 (config) #interface Ethernet 0/0
cisco3600 (config-if) #crypto map cisco2600
cisco3600 (config-if) #exit
cisco3600 (config) #
cisco3600 (config) #exit
```

Abbildung 14: Konfigurieren der Router

4.10 Bandbreitenmessung mit und ohne VPN-Tunnel

Nachdem die StudentInnen den VPN-Tunnel aufgesetzt haben, können sie die Messung der Bandbreite mit und ohne Tunnel vergleichen. Da die Router die Verschlüsselung und die Entschlüsselung auf Softwarebasis durchführen, ist die maximale Bandbreite bei etabliertem Tunnel um fast den Faktor vier kleiner wie die folgenden Abbildungen (Abbildung 15 bis Abbildung 18) zeigen:

```
host1@host1:~$ NPtcp -l 23552 -u 27648 -i 128 -r
host1@host1:~$
```

Abbildung 15: : Bandbreitenmessung ohne VPN-Tunnel (Empfänger)

```
host3@host3:~$ NPtcp -l 23552 -u 27648 -i 128 -t -h host1 -P
Latency: 0.000386
Now starting main loop
 0:    23552 bytes    7 times -->    6.89 Mbps in 0.026085 sec
 1:    23680 bytes    9 times -->    6.64 Mbps in 0.027214 sec
 2:    23808 bytes    9 times -->    6.81 Mbps in 0.026657 sec
 3:    23936 bytes    9 times -->    6.90 Mbps in 0.026460 sec
 4:    24064 bytes    9 times -->    7.06 Mbps in 0.026020 sec
 5:    24192 bytes    9 times -->    6.84 Mbps in 0.026997 sec
 6:    24320 bytes    9 times -->    6.88 Mbps in 0.026987 sec
 7:    24448 bytes    9 times -->    7.29 Mbps in 0.025570 sec
 8:    24576 bytes    9 times -->    7.02 Mbps in 0.026710 sec
 9:    24704 bytes    9 times -->    6.51 Mbps in 0.028954 sec
10:    24832 bytes    8 times -->    7.28 Mbps in 0.026010 sec
11:    24960 bytes    9 times -->    6.68 Mbps in 0.028512 sec
12:    25088 bytes    8 times -->    6.89 Mbps in 0.027792 sec
13:    25216 bytes    8 times -->    6.88 Mbps in 0.027980 sec
14:    25344 bytes    8 times -->    7.03 Mbps in 0.027505 sec
15:    25472 bytes    9 times -->    6.80 Mbps in 0.028584 sec
16:    25600 bytes    8 times -->    6.81 Mbps in 0.028680 sec
17:    25728 bytes    8 times -->    6.73 Mbps in 0.029174 sec
18:    25856 bytes    8 times -->    6.90 Mbps in 0.028569 sec
19:    25984 bytes    8 times -->    6.87 Mbps in 0.028856 sec
20:    26112 bytes    8 times -->    7.00 Mbps in 0.028452 sec
21:    26240 bytes    8 times -->    6.80 Mbps in 0.029461 sec
22:    26368 bytes    8 times -->    6.58 Mbps in 0.030584 sec
23:    26496 bytes    8 times -->    6.63 Mbps in 0.030506 sec
24:    26624 bytes    8 times -->    6.71 Mbps in 0.030294 sec
25:    26752 bytes    8 times -->    6.73 Mbps in 0.030328 sec
26:    26880 bytes    8 times -->    6.76 Mbps in 0.030347 sec
27:    27008 bytes    8 times -->    6.80 Mbps in 0.030299 sec
28:    27136 bytes    8 times -->    6.85 Mbps in 0.030236 sec
29:    27264 bytes    8 times -->    6.90 Mbps in 0.030160 sec
30:    27392 bytes    8 times -->    6.96 Mbps in 0.030005 sec
31:    27520 bytes    8 times -->    6.61 Mbps in 0.031757 sec
32:    27648 bytes    7 times -->    6.84 Mbps in 0.030827 sec
host3@host3:~$
```

Abbildung 16: Bandbreitenmessung ohne VPN-Tunnel (Sender)

```
host1@host1:~$ NPtcp -l 23552 -u 27648 -i 128 -r
host1@host1:~$
```

Abbildung 17: Bandbreitenmessung mit VPN-Tunnel (Empfänger)

```
host3@host3:~$ NPtcp -l 23552 -u 27648 -i 128 -t -h host1 -P
Latency: 0.001588
Now starting main loop
 0:      23552 bytes    7 times -->    1.73 Mbps in 0.103630 sec
 1:      23680 bytes    7 times -->    1.73 Mbps in 0.104233 sec
 2:      23808 bytes    7 times -->    1.73 Mbps in 0.105057 sec
 3:      23936 bytes    7 times -->    1.73 Mbps in 0.105674 sec
 4:      24064 bytes    7 times -->    1.73 Mbps in 0.106145 sec
 5:      24192 bytes    7 times -->    1.73 Mbps in 0.106635 sec
 6:      24320 bytes    7 times -->    1.73 Mbps in 0.107167 sec
 7:      24448 bytes    7 times -->    1.73 Mbps in 0.107797 sec
 8:      24576 bytes    7 times -->    1.73 Mbps in 0.108448 sec
 9:      24704 bytes    7 times -->    1.72 Mbps in 0.109476 sec
10:      24832 bytes    7 times -->    1.73 Mbps in 0.109615 sec
11:      24960 bytes    7 times -->    1.74 Mbps in 0.109712 sec
12:      25088 bytes    7 times -->    1.74 Mbps in 0.110256 sec
13:      25216 bytes    7 times -->    1.74 Mbps in 0.110845 sec
14:      25344 bytes    7 times -->    1.73 Mbps in 0.111461 sec
15:      25472 bytes    7 times -->    1.74 Mbps in 0.111942 sec
16:      25600 bytes    7 times -->    1.74 Mbps in 0.112439 sec
17:      25728 bytes    7 times -->    1.74 Mbps in 0.112902 sec
18:      25856 bytes    7 times -->    1.74 Mbps in 0.113520 sec
19:      25984 bytes    7 times -->    1.74 Mbps in 0.113992 sec
20:      26112 bytes    7 times -->    1.73 Mbps in 0.115106 sec
21:      26240 bytes    7 times -->    1.74 Mbps in 0.115245 sec
22:      26368 bytes    7 times -->    1.74 Mbps in 0.115393 sec
23:      26496 bytes    7 times -->    1.74 Mbps in 0.115969 sec
24:      26624 bytes    7 times -->    1.74 Mbps in 0.116748 sec
25:      26752 bytes    7 times -->    1.74 Mbps in 0.117277 sec
26:      26880 bytes    7 times -->    1.74 Mbps in 0.117776 sec
27:      27008 bytes    7 times -->    1.74 Mbps in 0.118310 sec
28:      27136 bytes    7 times -->    1.74 Mbps in 0.118795 sec
29:      27264 bytes    7 times -->    1.74 Mbps in 0.119295 sec
30:      27392 bytes    7 times -->    1.74 Mbps in 0.120023 sec
31:      27520 bytes    7 times -->    1.73 Mbps in 0.121303 sec
32:      27648 bytes    7 times -->    1.74 Mbps in 0.121452 sec
host3@host3:~$
```

Abbildung 18: Bandbreitenmessung mit VPN-Tunnel (Sender)

4.11 Mitlesen (“Sniffen”) von Passwörtern mit und ohne VPN-Tunnel

Um sich zu versichern, dass die übertragenen Daten durch den VPN-Tunnel vor dem Mitlesen durch Drittpersonen geschützt sind, lesen die StudentInnen den Netzwerkverkehr zwischen den Linux-Rechnern host1 und host3 mittels des Werkzeuges Tcpdump (Abbildung 19) mit. Der zu testende Verkehr wird mittels des Programms Telnet (Abbildung 20) erzeugt, welches den Anmeldenamen und das Passwort im Klartext über das Netzwerk überträgt. Als Anmeldenamen wurde „ABC“ verwendet, damit dessen Klartext (Hexadezimal 414243) in der Ausgabe von Tcpdump leichter gefunden wird. Die folgenden Abbildungen (Abbildung 21 und Abbildung 22) illustrieren die Unterschiede. Die relevanten Daten wurden fett und grösser markiert.

```
bash-2.05a$ tcpdump -i eth2 -x 'port 23 and src host host1 and dst host host3'
```

Abbildung 19: Tcpdump: Mitlesen des Verkehrs vom Rechner host1 zum Rechner host3

```
host1@host1:~$ telnet host3
Trying 10.3.0.100...
Connected to host3.
Escape character is '^]'.
Debian GNU/Linux 2.2 host3
host3 login: ABC
```

Abbildung 20: Telnet vom host1 zum host3

```
bash-2.05a$ tcpdump -i eth2 -x 'port 23 and src host host1 and dst host host3'
tcpdump: listening on eth2
11:21:32.397054 host1.1127 > host3.23: P 4291357857:4291357858(1) ack 4177226265 win 32120 <nop,nop,timestamp 215896279 215890988> (DF) [tos 0x10]
    4510 0035 7dc9 4000 3f06 a91e 0a01 0064
    0a03 0064 0467 0017 ffc8 eca1 f8fb 6a19
    8018 7d78 a8b2 0000 0101 080a 0cde 50d7
    0cde 3c2c 41
11:21:32.417339 host1.1127 > host3.23: . ack 2 win 32120
<nop,nop,timestamp 215896282 215892548> (DF) [tos 0x10]
    4510 0034 7dcb 4000 3f06 a91d 0a01 0064
    0a03 0064 0467 0017 ffc8 eca2 f8fb 6a1a
    8010 7d78 e39e 0000 0101 080a 0cde 50da
    0cde 4244
11:21:33.369370 host1.1127 > host3.23: P 1:2(1) ack 2 win 32120
<nop,nop,timestamp 215896377 215892548> (DF) [tos 0x10]
    4510 0035 7dcc 4000 3f06 a91b 0a01 0064
    0a03 0064 0467 0017 ffc8 eca2 f8fb 6a1a
    8018 7d78 a136 0000 0101 080a 0cde 5139
    0cde 4244 42
11:21:33.387384 host1.1127 > host3.23: . ack 3 win 32120
```



```

<nop,nop,timestamp 215896379 215892646> (DF) [tos 0x10]
    4510 0034 7dce 4000 3f06 a91a 0a01 0064
    0a03 0064 0467 0017 ffc8 eca3 f8fb 6a1b
    8010 7d78 e2d9 0000 0101 080a 0cde 513b
    0cde 42a6
11:21:34.374722 host1.1127 > host3.23: P 2:3(1) ack 3 win 32120
<nop,nop,timestamp 215896477 215892646> (DF) [tos 0x10]
    4510 0035 7dcf 4000 3f06 a918 0a01 0064
    0a03 0064 0467 0017 ffc8 eca3 f8fb 6a1b
    8018 7d78 9f6e 0000 0101 080a 0cde 519d
    0cde 42a6 43
11:21:34.387396 host1.1127 > host3.23: . ack 4 win 32120
<nop,nop,timestamp 215896479 215892746> (DF) [tos 0x10]
    4510 0034 7ddl 4000 3f06 a917 0a01 0064
    0a03 0064 0467 0017 ffc8 eca4 f8fb 6a1c
    8010 7d78 e20f 0000 0101 080a 0cde 519f
    0cde 430a

6 packets received by filter
0 packets dropped by kernel
bash-2.05a$

```

Abbildung 21: Tcpcdump-Ausgabe ohne VPN-Tunnel

```

bash-2.05a$ tcpdump -i eth2 -x 'port 23 and src host host1 and dst
host host3'
tcpdump: listening on eth2
12:06:59.819850 host1.1131 > host3.23: P 2889459632:2889459633(1) ack
2745856051 win 32120 <nop,nop,timestamp 216169014 216164396> (DF)
[tos 0x10]
    4510 0035 7e7c 4000 3f06 a86b 0a01 0064
    0a03 0064 046b 0017 ac39 a7b0 a3aa 7033
    8018 7d78 3aff 0000 0101 080a 0ce2 7a36
    0ce2 682c 69
12:06:59.837620 host1.1131 > host3.23: . ack 2 win 32120
<nop,nop,timestamp 216169016 216165281> (DF) [tos 0x10]
    4510 0034 7e7e 4000 3f06 a86a 0a01 0064
    0a03 0064 046b 0017 ac39 a7b1 a3aa 7034
    8010 7d78 788f 0000 0101 080a 0ce2 7a38
    0ce2 6ba1
12:07:02.330406 host1.1131 > host3.23: P 1:2(1) ack 2 win 32120
<nop,nop,timestamp 216169265 216165281> (DF) [tos 0x10]
    4510 0035 7e7f 4000 3f06 a868 0a01 0064
    0a03 0064 046b 0017 ac39 a7b1 a3aa 7034
    8018 7d78 358d 0000 0101 080a 0ce2 7b31
    0ce2 6ba1 44
12:07:02.347619 host1.1131 > host3.23: . ack 3 win 32120
<nop,nop,timestamp 216169267 216165532> (DF) [tos 0x10]
    4510 0034 7e81 4000 3f06 a867 0a01 0064
    0a03 0064 046b 0017 ac39 a7b2 a3aa 7035
    8010 7d78 7697 0000 0101 080a 0ce2 7b33
    0ce2 6c9c
12:07:04.487380 host1.1131 > host3.23: P 2:3(1) ack 3 win 32120
<nop,nop,timestamp 216169480 216165532> (DF) [tos 0x10]
    4510 0035 7e82 4000 3f06 a865 0a01 0064
    0a03 0064 046b 0017 ac39 a7b2 a3aa 7035
    8018 7d78 32b9 0000 0101 080a 0ce2 7c08
    0ce2 6c9c 4e
12:07:04.507726 host1.1131 > host3.23: . ack 4 win 32120
<nop,nop,timestamp 216169483 216165748> (DF) [tos 0x10]

```

```
4510 0034 7e84 4000 3f06 a864 0a01 0064
0a03 0064 046b 0017 ac39 a7b3 a3aa 7036
8010 7d78 74e5 0000 0101 080a 0ce2 7c0b
0ce2 6d74

6 packets received by filter
0 packets dropped by kernel
bash-2.05a$
```

Abbildung 22: Tcpdump-Ausgabe mit VPN-Tunnel

5 Die VITELS-Architektur

Das im Kapitel 6 beschriebene Internetportal ist Teil einer am IAM entwickelten und in die Praxis umgesetzten verteilten Lernarchitektur, der VITELS-Architektur [5][56][57]. Diese erlaubt es, örtlich-verteilte Institutionen so zu verbinden, dass eine gemeinsame verteilte Lernumgebung entsteht. Wie in Abbildung 23 dargestellt, besteht sie aus den folgenden Elementen:

- **Kursserver**
Auf dem Kursserver läuft diejenige Applikation, die die einzelnen Kursmodule in eine einheitliche Umgebung zusammenfasst. Sie bietet den StudentInnen Theorie, Übungen, Hilfe und die Möglichkeit sich mittels Diskussionsforen, Whiteboards, Listen von häufig gestellten Fragen und Chaträumen mit anderen StudentInnen auszutauschen. Der Kursserver ist der Haupteinstiegspunkt für die StudentInnen.
- **Studentendatenbank(en)**
In den Studentendatenbanken sind die Attribute der StudentInnen gespeichert. Diese Attribute enthalten Informationen über die StudentInnen, wie z.B. Vornamen, Nachnamen, Benutzernamen, Passwörter, Matrikelnummern, Emailadressen, etc.
- **Internetportal(e)**
Das Internetportal ist die Schnittstelle zwischen dem StudentInnen und dem dahinter liegenden Kursmodul bzw. den Laborgeräten. Am Portal melden sich die StudentInnen an, um Zugriff auf die Laborgeräte zu erhalten.
- **Kursmodul(e)**
Das Kursmodul besteht aus den Laborgeräten, die den StudentInnen zwecks Konfiguration, Messung und Übungen zur Verfügung gestellt werden. Die Kursmodule des VITELS-Basiskurses sind in der Tabelle 1 aufgeführt. Eines dieser Kursmodule ist das Modul IPSecurity, welches aufgrund eines traditionellen Computernetze-Praktikumversuchs (siehe auch Kapitel 4) im Rahmen dieser Arbeit für den Fernzugriff angepasst wurde. Die Details dieser Anpassung befinden sich im Kapitel 7.
- **Administration**
Von hier aus greifen DozentInnen auf den Kursserver zu und kontrollieren die Leistungen der StudentInnen oder erfassen und / oder mutieren Studentendaten in den Studentendatenbank(en). Sie benützen hierzu ihren Arbeitsrechner und einen Browser, um mittels des Webfrontends des Kurssystems WebCT (siehe auch Abbildung 25) auf die Studentendaten zuzugreifen.

- **Netzwerkverbindungen**

Sie verbinden die oben beschriebenen Komponenten mittels sicheren Kommunikationskanälen wie z.B. IPSec, SSH oder HTTPS um die sensiblen Studentendaten zu schützen. Verbindungen zwischen Kursserver, Studentendatenbank(en) und Internetportal(en) werden mittels IPSec gesichert. Über all dort, wo Webfrontends verwendet werden, wird die Kommunikation durch HTTPS verschlüsselt, während Anmeldungen der StudentInnen am Internetportal mit SSH übertragen werden.

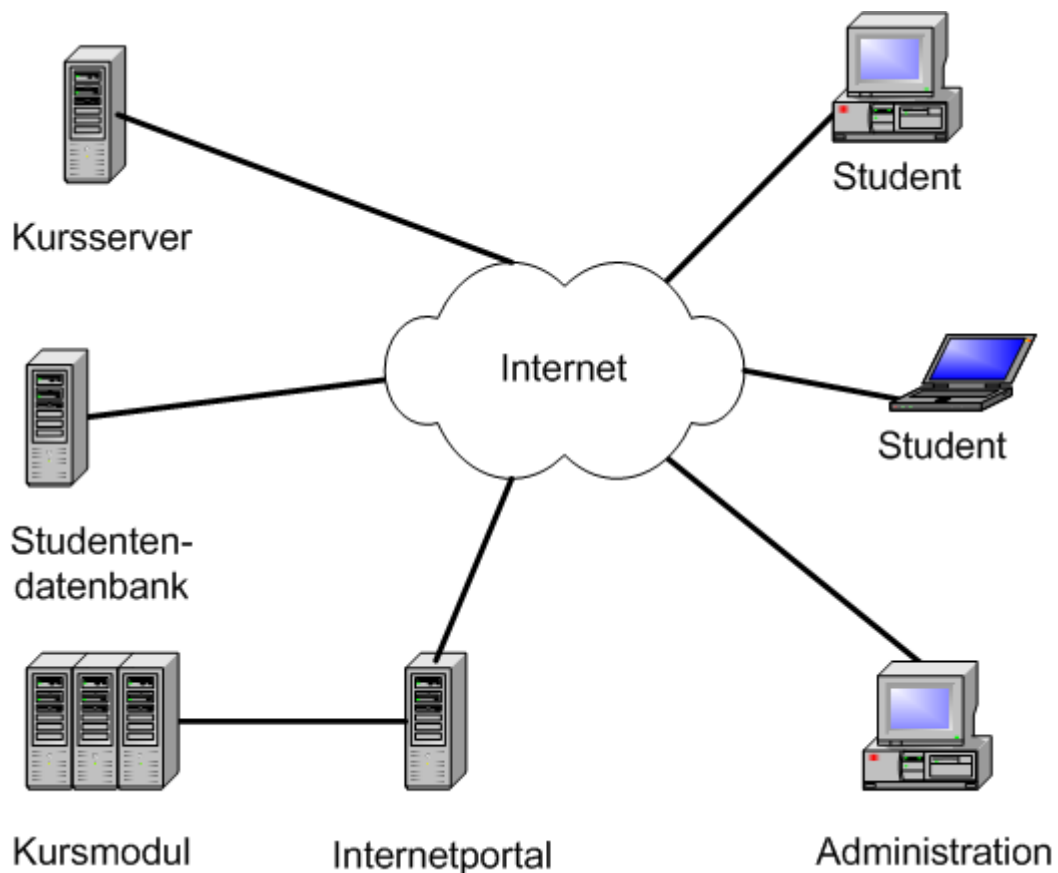


Abbildung 23: Die VITELS-Architektur

5.1 Kursserver

Der Kursserver basiert auf dem im Kapitel 2.3.2.1 beschriebenen Kurssystem WebCT und wird – wie die Studentendatenbank - von den Informatikdiensten der Universität Bern betrieben. Obwohl durch die Architektur nicht verlangt, sind das Kurssystem und die Studentendatenbank auf dem gleichen Rechner installiert. Nach dem erfolgreichen Anmelden präsentiert sich die Auswahl der Module und das Reservierungssystem auf dem Kurssystem den StudentInnen gemäss folgender Abbildung 24.

The screenshot shows the WebCT interface for the Virtual Internet and Telecommunications Laboratory of Switzerland (VITELS). The browser title is "Virtual Internet and Telecommunications Laboratory of Switzerland VITELS - WebCT 3.6.3 - Mozilla (build ID: 2002053012)". The page header includes navigation links: MYWEBCT | RESUME COURSE | COURSE MAP | RESOURCES | HELP. The main content area displays the course title and a message: "Work on real hardware, at real-time! Please select the module you would like to attend. We propose to work from module 1 to 7." Below this, seven modules are listed with icons and links:

- 1) Linux Systems Installation and Configuration
- 2) Simulation of IP Network Configuration
- 3) Configuration and Performance Evaluation of a Real IP Network (Hidden)
- 4) Client/Server Programming (Hidden)
- 5) Protocol Analysis (Hidden)
- 6) IP Security
- 7) Firewall Management

At the bottom right, there is a reservation calendar for Module 1, 6, 7. The calendar is a grid with columns for days of the week (MON, TUE, WED, THU, FRI, SAT) and rows for time slots. A yellow box labeled "RECESS" covers the bottom portion of the calendar. Mickey Mouse is on the left and Donald Duck is on the right of the calendar.

Abbildung 24: VITELS-Kursmodule und Modulreservierung im Kurssystem WebCT

Virtual Internet and Telecommunications Laboratory of Switzerland VITELS - WebCT 3.6.3 - Mozilla (Build ID: 2002053012)

File Edit View Go Bookmarks Tools Window Help

WebCT

MYWEBCT | RESUME COURSE | COURSE MAP | RESOURCES | HELP

Hide Navigation

Manage Students: Designer Options

Virtual Internet and Telecommunications Laboratory of Switzerland VITELS

Home » Designer Map » Update Student View » Track Students » Manage Students

Manage Students **Advanced Options**

Select action Select action

Page: Displaying records 1 - 65 of 65 [Total: 65]

First Name	Last Name	User ID	Email	Fakultaet	Matrikelnummer	Linux Installation and Configuration 2	simlPconff: know
Sort Edit	Sort Edit	Sort	Sort Edit	Sort Edit	Sort Edit	Sort Submissions Graph Out of 6	Sort Submissio Graph Out of 14
Abdel kader	Fall	akfall	fall1@etu.unige.ch	unige	---	---	---
Abdelmajid	Jabri	ajabri	jabri9@etu.unige.ch	unige	---	---	---
Alfredo	Villalba	avillalba	villalb0@etu.unige.ch	unige	---	---	---
Anna	Siegenthaler	asiegen+	anna.siegenthaler@d.unibe.ch	---	---	---	---
Asheesh	Gulati	agulati	gulati9@etu.unige.ch	unige	---	---	---
Basilio	Noris	bnoris	dies-irae@dies-irae.com	---	---	---	---
Christiane	James	cjames	james99@cunmail.unige.ch	unige	---	---	---
Christine	Rosenberger	rosenber?	rosenber@student.unibe.ch	phil.nat	89103683	---	---
Cynthia	Schaller	cschaller	schalle8@etu.unige.ch	unige	---	---	---
Cyril	Marti	cm98c076	cm98c076@student.unibe.ch	phil.nat	98113533	---	---
Damantang	Camara	dcamara	camardb0@etu.unige.ch	unige	---	---	---
Daniel	Balsiger	db00a000	db00a000@student.unibe.ch	phil.nat	98908213	---	---
Didier	Baertschiger	dbaertschiger	baertsc0@etu.unige.ch	unige	---	---	---
Duy	Nguyen	dnguyen	nguyend2@etu.unige.ch	unige	---	---	---
El hadji abdoul aziz	Mbaye	mbaye	mbayee11@etu.unige.ch	unige	---	---	---
Elton	Rezhepaj	erezhepaj	elton.rezhepaj@etu.unil.ch	unil	---	---	---
Enea	Pestelacci	epestelacci	enea.pestelacci@unil.ch	unil	---	---	---
Ernesto	Rivera	erivera	rivear0@etu.unige.ch	unige	---	---	---
Etienne	Dysli	edysli	etienne.dysli@etu.unil.ch	unil	---	---	---
Gent	Bajrami	gbajrami	bajrami9@etu.unige.ch	unige	---	---	---
Gerda	Cabej	gcabej	cabej0@etu.unige.ch	unige	---	---	---
Giles	Rosset	grosset	rossetg0@etu.unige.ch	unige	---	---	---
Hilda	Hysi	hhysi	hysi9@etu.unige.ch	unige	---	---	---
Isabelle	Flückiger	iflueckiger	Isabelle.Flueckiger@etu.unil.ch	unil	---	---	---

Document: Done (6.269 secs)

Abbildung 25: Administrationsansicht der Studentendaten im Kurssystem WebCT

5.2 Studentendatenbank

Die Studentendatenbank wurde auf der Basis eines LDAP-Verzeichnisses implementiert und wird produktiv von den Informatikdiensten [58] der Universität Bern betrieben. Die zu Grunde liegenden Datenstrukturen werden im folgenden kurz beschrieben:

5.2.1 Verzeichnisstruktur für Universitäten

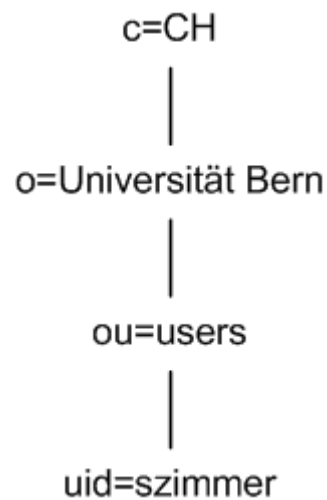


Abbildung 26: Verzeichnisstruktur für Universitäten

Abbildung 26 zeigt die Verzeichnisstruktur für Universitäten. Sie ist hierarchisch nach Land, Organisation, Unterorganisation und Individuum aufgebaut. Mittels dieser Struktur können Studentendaten in einem LDAP-Verzeichnis abgelegt werden.

5.2.2 Verzeichnisstruktur für VITELS-spezifische Einträge

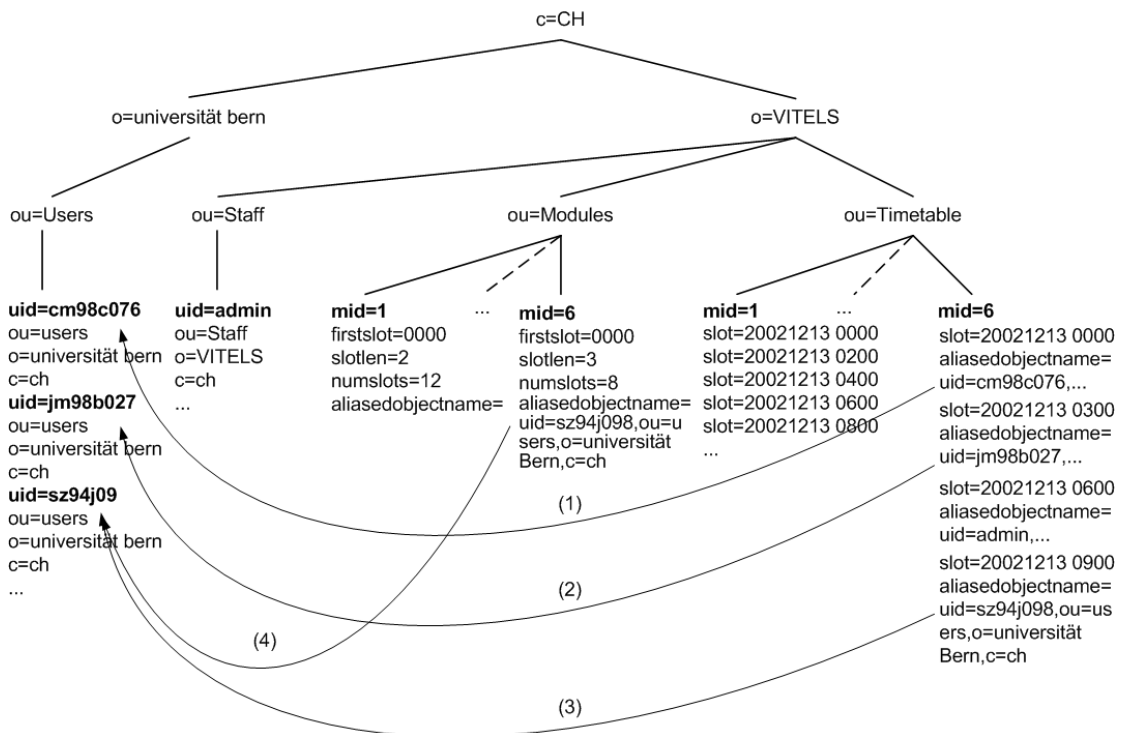


Abbildung 27: Verzeichnisstruktur für VITELS-spezifische Einträge

Abbildung 27 zeigt die Struktur für die VITELS-Einträge. Ganz links befinden sich die im Kapitel 5.2.1 erwähnten Studenteneinträge. Der Teilbaum Staff enthält gleichaufgebaute Einträge wie die der Studentendaten, aber die darin aufgenommenen Personen haben weiterreichende Rechte als die StudentInnen und administrieren einzelne Kursmodule.

Der Teilbaum Modules im VITELS-Ast enthält Kursmodulspezifische Daten wie z.B. die Modulnummer. Für jedes Kursmodul existiert ein Eintrag. Falls das Modul nicht mehr als einen Benutzer gleichzeitig zulässt, werden hier die für das Reservationssystem benötigten Daten, wie Länge (slotlen), Anzahl (numslots) und Startzeit (firstslot) der möglichen Zeitschlitze sowie der momentan eingetragene Benutzer (aliasedobjectname) gespeichert. Der Eintrag aliasedobjectname (siehe auch (4) in Abbildung 27) ist ein Zeiger (Alias) auf einen Datensatz der Studentendaten und legt den zur Zeit für das Kursmodul aktiven Benutzer fest. Der dritte Teilbaum Timetable enthält für jedes Modul Einträge für das Reservationssystem. Diese werden im folgenden Kapitel 5.3 beschrieben.

5.3 Reservierungssystem

Um Laborgeräte, die nicht für die gleichzeitige Benutzung durch mehrere StudentInnen vorgesehen sind, in den VITELS-Kurs integrieren zu können, wurde ein leistungsfähiges Reservierungssystem, welches Zugriff auf das LDAP-Verzeichnis hat, implementiert [5]. Die StudentInnen greifen über in PHP geschriebene Webseiten (Abbildung 28) darauf zu und können vom Moduladministrator vordefinierte Zeitschlitze (Timeslots) reservieren und freigeben. Sobald ein Student oder eine Studentin ein Modul reserviert hat, hat er oder sie für die Dauer des Timeslots exklusiven Zugriff auf das Modul. Zu diesem Zweck enthält der Teilbaum ou=Timetable für jedes Modul eine Liste von Timeslot-Einträgen, welche die verfügbaren Timeslots bezeichnen. Die hierzu definierten Attribute werden nachfolgend beschrieben.

Virtual Internet and Telecommunications Laboratory of Switzerland VITELS - WebCT 3.6.3 - Mozilla (Build ID: 2002033012)

File Edit View Go Bookmarks Tools Window Help

WebCT MYWEBCT | RESUME COURSE | COURSE MAP | RESOURCES | HELP

Hide Navigation Lab Reservation for Module: 1, 6, 7 View

Virtual Internet and Telecommunications Laboratory of Switzerland VITELS

Home - Lab Reservation for Module: 1, 6, 7

Timetable

1a 1b 1c 1d 2 3 5 6 7

previous week December 09 - December 15, 2002 next week

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
00:00 - 03:00	✓	○	○	○	○	✗	✗
03:00 - 06:00	○	○	○	○	○	✗	✗
06:00 - 09:00	✗	○	○	✗	✗	✗	✗
09:00 - 12:00	✗	○	✗	✗	✗	✗	✗
12:00 - 15:00	○	✗	✗	✗	○	✗	✗
15:00 - 18:00	○	✗	✗	✗	✗	✗	✓
18:00 - 21:00	○	✗	✗	✗	✗	✓	✗
21:00 - 00:00	○	✗	✗	✗	✗	✗	✗

○ slot free ✓ your slot ✗ slot reserved

Document: Done (0.971 secs)

Abbildung 28: Das Reservationssystem

5.3.1 Attribute der Timetable-Einträge

Attribut	Wert
ou	Timetable
mid	6

Tabelle 4: Attribute der Timetable-Einträge

Für jedes VITELS-Kursmodul existiert ein solcher Eintrag im VITELS-Ast. Das Attribut mid legt fest, zu welchem Modul dieser Eintrag gehört. Mit den Werten in Tabelle 4 wird definiert, dass diese Timetable dem Modul 6 zugeordnet ist.

Unterhalb der Timetable-Einträge (siehe auch Abbildung 27) befinden sich die Timeslot-Einträge, welche die für die Benutzer verfügbaren Zeitschlitze des jeweiligen Moduls definieren. Pro Zeitschlitz existiert ein Timeslot-Eintrag. Diese werden durch den Moduladministrator erzeugt. Im Attribut mid steht die Modulnummer, in slot der Beginn des Timeslots. Das Attribut aliasedobjectname ist nach dem Erzeugen der Timeslots leer. Reserviert sich ein Student oder eine Studentin einen Timeslot, wird in aliasedobjectname ein Zeiger auf dessen Eintrag in den Studentendaten erzeugt. In der Abbildung 27 sind vier solcher Reservationseinträge im Modul 6 gezeichnet. Die nummerierten Pfeile (1), (2) und (3) zeigen die Aliase, die dadurch erzeugt werden. Die folgende Tabelle 5 zeigt einen dieser Einträge:

Attribut	Wert
ou	Timetable
mid	6
slot	20021213 0900
aliasedobjectname	uid=sz94j098,ou=users,o=Universität Bern,c=ch

Tabelle 5: Attribute der Timeslot-Einträge

Somit wird ein Timeslot im Modul 6 definiert, welcher am 13.12.2002 um 9 Uhr 00 beginnt. Der Timeslot wurde durch den Benutzer mit der eindeutigen Kennung sz94j098 reserviert.

5.3.2 Attribute der Module-Einträge

Die Moduleinträge befinden sich im VITELS-Ast und es existiert pro Kursmodul ein Eintrag mit den Attributen der folgenden Tabelle 6:

Attribut	Wert
mid	6
firstslot	0000
slotlen	3
numslots	8
aliasedobjectname	uid=sz94j098,ou=users,o=Universität Bern,c=ch

Tabelle 6: Attribute der Module-Einträge

Mittels diesen Werten wird für das Modul 6 festgelegt, dass die Timeslots 3 Stunden dauern, pro Tag 8 Timeslots zur Verfügung stehen und der erste Slot um Mitternacht beginnt. Durch das letzte Attribut `aliasedobjectname` wird der momentan aktive Benutzer, der sogenannte „Current User“ des Moduls angegeben. Der Inhalt dieses Attributes wird durch den Wert des aktuellen Timeslots aus der Timetable definiert. Gemäss dem Beispiel in der Abbildung 27 heisst dies, dass am 13.12.2002 von Mitternacht bis um 3 Uhr 00 die Kennung des aktiven Benutzers `cm98c076` ist, von 3 Uhr 00 bis um 6 Uhr 00 ist es `jm98b027`, von 6 Uhr 00 bis um 9 Uhr 00 ist es `admin` und danach die Kennung `sz94j098`.

Auf Grund dieser Information kann festgestellt werden, wer zur aktuellen Zeit das Modul reserviert hat.

6 Internetportal

In diesem Kapitel wird auf das Konzept und die Realisierung des Internetportals als Teilkomponente eines verteilten Lernsystems eingegangen. Zuvor werden die an das Portal gestellten Anforderungen ausgeführt.

6.1 Anforderungen an das Internetportal

Die Anforderungen an das Internetportal werden in die folgenden Teilbereiche aufgeteilt, die jeweils eine bestimmte Schnittstelle zu einer der Komponenten der im Kapitel 5 beschriebenen Architektur haben:

- **Schnittstelle zu den Laborgeräten**
Diese Schnittstellen ermöglichen es, die Geräte eines Kursmoduls mit dem Portal zu verbinden. Dies können Netzwerkkarten, serielle oder parallele Schnittstellen sein. Mittels dieser Schnittstellen werden die Laborgeräte durch das Portal überwacht und konfiguriert.
- **Schnittstelle zu den StudentInnen**
Dies ist der Einstiegspunkt für die StudentInnen. Über sie präsentiert sich das Fernkurslabor dem Benutzer und kann z.B. mittels Webseiten realisiert werden. An dieser Schnittstelle muss sich der Benutzer mit seinem Namen und einem Passwort anmelden.
- **Schnittstelle zur Benutzerdatenbank**
Mittels dieser Schnittstelle ist das Portal in der Lage zu entscheiden, ob die Informationen, die es über die Studentenschnittstelle erhalten hat, den oder die Studentin berechtigen, auf das Labor zu zugreifen. Da über diesen Kanal sensible private Daten ausgetauscht werden, muss dieser die entsprechenden Vorkehrungen treffen, damit diese Daten geschützt werden können. Dies kann durch eine dezidierte private Leitung oder starke Verschlüsselung erreicht werden.
- **Schnittstelle zum Kursserver**
Der Kursserver führt durch den Kurs und integriert die einzelnen Module in eine einheitliche Umgebung. Via diese Schnittstelle greift der Kursserver auf das Portal zu und leitet den oder die gerade aktiven Benutzer an das Portal um. Zusätzlich können über diesen Kanal Informationen, die zur Leistungsbewertung der StudentInnen dienen, an den Kursserver zurückgeschickt werden.

- **Schnittstelle zum Reservierungssystem**
Sind die verfügbaren Labor-Ressourcen beschränkt, so kann das Portal über diese Schnittstelle nachprüfen, ob der ankommende Benutzer das Labor für die aktuelle Zeit reserviert hat oder nicht.
- **Schnittstelle zum Abrechnungssystem**
Sollen der Gebrauch der Laborgeräte in Rechnung gestellt werden, können über diese Schnittstelle die entsprechenden Daten in ein Abrechnungssystem, ein Clearinghouse eingespielen werden.

6.2 Das Konzept des Internetportals

Wie im vorigen Kapitel beschrieben, werden diverse Anforderungen an das Internetportal gestellt. Um diesen Anforderungen gerecht zu werden, wurde ein Konzept erstellt, welches aus drei Hauptmerkmalen besteht:

- Das Portal als Hub
- Das Portal als Firewall
- Abbildung der Portaluser auf die Laborgeräte

Diese Eigenschaften werden in den folgenden Abschnitten beschrieben.

6.2.1 Das Portal als Hub

Das Portal besitzt mehrere Schnittstellen zu den umliegenden Systemen wie Benutzerdatenbank, Reservationssystem und StudentIn. Es kommuniziert mit diesen über mehrere Kanäle und verbindet die StudentInnen mit den Laborgeräten. Aus diesem Grund wird es als Schnittstellenkonzentrator oder Hub aufgefasst wie in der Abbildung 29 dargestellt ist.

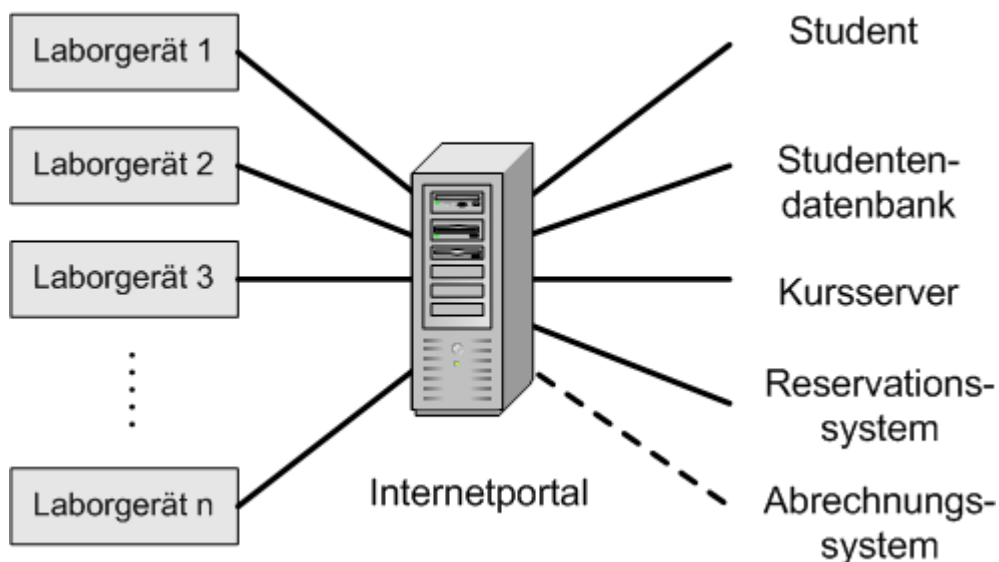


Abbildung 29: Das Portal als Hub

6.2.2 Das Portal als Firewall

Aufgrund der Schnittstellenvielfalt des Portals und dessen Fähigkeit, zu entscheiden, ob Daten von einer Schnittstelle A kommend, über eine Schnittstelle B an ein anderes System weitergereicht werden, besitzt das Portal Firewall-Funktionalität. Beispielsweise (Abbildung 30) sei hier genannt, dass die StudentInnen, die auf das Portal zugreifen, über dieses keinen Zugriff auf die Studentendatenbank haben, die ja auch mit dem Portal verbunden ist.

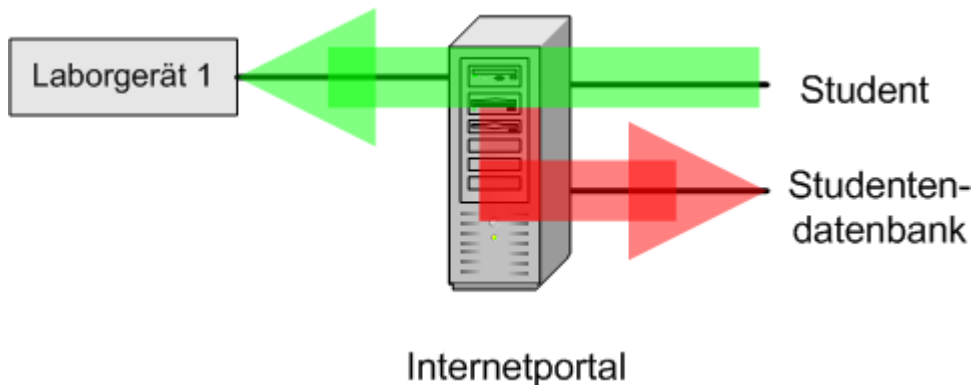


Abbildung 30: Das Portal als Firewall

6.2.3 Abbildung der Portalbenutzer auf Laborgeräte

Die Laborgeräte müssen vom Portal aus angesprochen, überwacht und konfiguriert werden können. Weil die StudentInnen via Portal auf die Laborgeräte zugreifen, ist es naheliegend, dass sich die Laborbenutzer am Portal anmelden und von diesem authentifiziert werden müssen. Dies wird erreicht, indem für jedes Gerät, das an das Portal angeschlossen wird, ein eigener Benutzer (ein User) auf Betriebssystemebene des Portals definiert wird. Betriebssystem-User wurden daher ausgewählt, da diese durch Skripte relativ bequem angelegt, verändert und gelöscht werden können. Weiter können sie gegen nahezu beliebig viele Benutzerdatenbanken authentifiziert werden (siehe Kapitel 6.2.4).

6.2.4 Authentifizierungsmethoden

6.2.4.1 Authentifizierung mittels lokalen Benutzer- / Passwortlisten

Bei dieser Methode werden Benutzer gegen die zwei Linux-Systemdateien `passwd` und `shadow` authentifiziert. Die Datei `passwd` enthält den eindeutigen Benutzernamen (das Login), sowie weitere Informationen, wie Vornamen, Nachnamen und die Gruppenzugehörigkeit des Benutzers. In der Datei `shadow` wird das Benutzerpasswort verschlüsselt abgespeichert, so dass es praktisch unmöglich ist, vom verschlüsselten Passwort Rückschlüsse auf dessen Klartext zu ziehen. Diese Methode empfiehlt sich für sehr kleine Netzwerke, da jeder Benutzer auf jedem System angelegt werden muss.

6.2.4.2 Authentifizierung mittels des Network Information Service (NIS)

Das Network Information System (NIS), früher Yellow Pages (YP) genannt, wurde von Sun Microsystems [64] entwickelt und basiert auf dem Client/Server-Prinzip. Jeder Rechner im Netzwerk der als NIS-Client konfiguriert ist, hat Zugriff auf die Benutzerdaten, die zentral auf dem NIS-Server gespeichert sind. NIS benützt Remote Procedure Calls (RPC) [65] und bietet den Vorteil, dass es für einen Benutzer keine Rolle spielt, ob er sich am Rechner A oder Rechner B anmeldet.

6.2.4.3 Authentifizierung mittels LDAP

Sind die Benutzerdaten in einem zentralen LDAP-Verzeichnis gespeichert, können Rechner die Benutzer gegen dieses Verzeichnis authentifizieren. Hierzu wurde das Modul pam_ldap [66] entwickelt, welches das Pluggable Authentication Module (PAM) [67] benutzt.

In dieser Arbeit wird die Authentifizierung der Systemuser und damit der Zugriff auf die Laborgeräte mittels lokaler Benutzer- / Passwortlisten durchgeführt. Dessen Realisierung wird im folgenden Abschnitt beschrieben.

6.3 Realisierung

Mittels des im vorigen Kapitel 6.2 beschriebenen Konzepts wurde das Internetportal realisiert und in die VITELS-Architektur eingebettet. Die dafür realisierten Anbindungen werden durch die folgende Abbildung 31 illustriert:

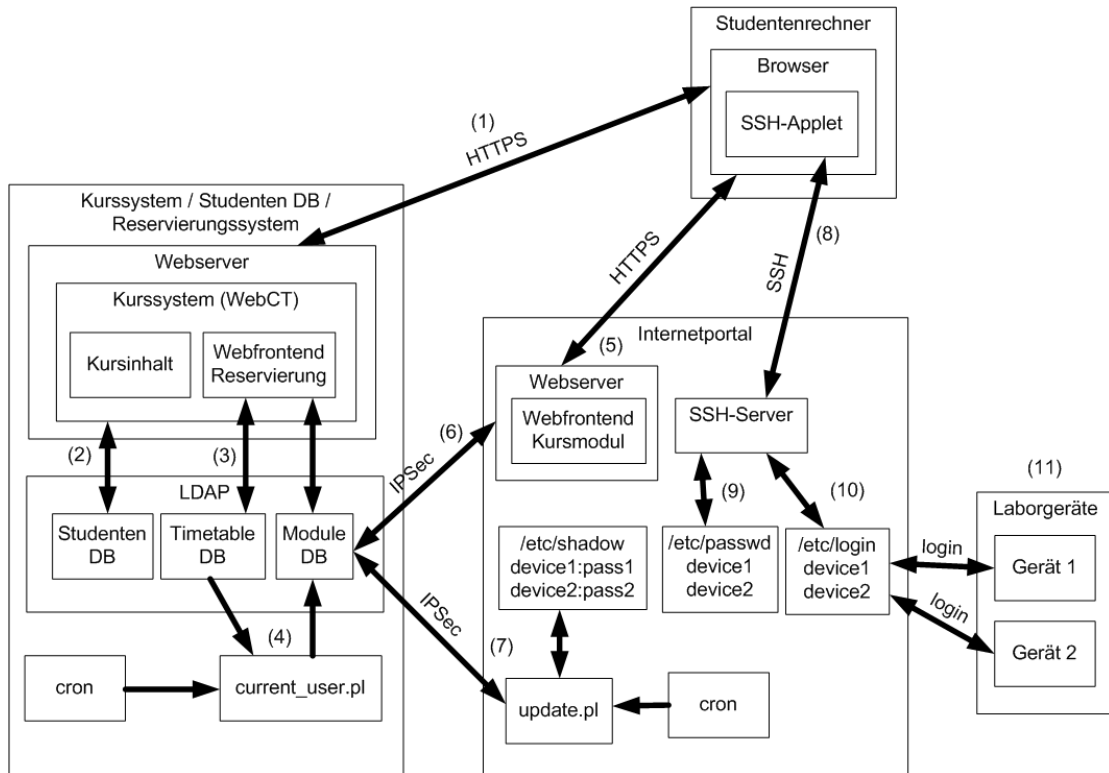


Abbildung 31: Anbindung des Internetportals an die VITELS-Architektur

6.3.1 Ablauf der Anmeldung

In diesem Abschnitt wird anhand der Abbildung 31 beschrieben, wie der gesamte Anmeldevorgang vollzogen wird:

- 1) Zuerst geht der Student mit dem Browser auf die Webseite des Kurssystems und meldet sich mittels seinem Benutzernamen und Passwort an.
- 2) Das Kurssystem prüft dies, indem es auf die Studentendaten im LDAP-Verzeichnis zugreift. Die Verbindung zwischen dem Browser und dem Webserver ist durch HTTPS geschützt.
- 3) Nach erfolgreicher Anmeldung präsentiert ihm das Kurssystem die Hauptseite, mittels welcher er sich einen Timeslot reservieren kann, welcher im LDAP-Verzeichnis in der Timetable-Datenbank als besetzt markiert wird.

- 4) Das Skript „current_user.pl“ (siehe auch [5], Kapitel 4.3.1) auf dem Kurs- / Reservierungssystem wird durch einen CRON-Job einmal pro Minute gestartet, liest das Benutzerattribut des aktuellen Timeslots-Eintrags im Teilbaum Timetable und schreibt diesen als „Current User“ in das Benutzerattribut (siehe auch Abbildung 27) des ausgewählten Moduls. Dieses Modul ist somit für den Studenten oder die Studentin reserviert.
- 5) Klickt der Student im Kurssystem auf den Link, der zum Internetportal führt, öffnet sein Browser das Webfrontend (siehe auch Kapitel 6.3.2) des Kursmoduls. Auch diese Verbindung wird durch HTTPS gesichert. Der Student muss sich am Kursmodul nochmals anmelden, da es nicht möglich ist, die am Kurssystem bereits erfolgte Anmeldung an das Kursmodul weiterzugeben.
- 6) Der im Internetportal integrierte Webserver greift nun über eine durch IPsec geschützte Netzwerkverbindung auf die Moduldaten im LDAP-Verzeichnis zu und prüft, ob das angegebene Benutzerattribut und das Passwort demjenigen entsprechen, der momentan als „Current User“ für dieses Modul eingetragen ist. Dies wird mittels der VITELS-LDAP-Schnittstelle ([5], Kapitel 4.5) erreicht, welche bei Erfolg die Start- und Endzeit des reservierten Timeslots zurückgibt. Aufgrund dieser Information wird nun ein Sessioncookie erzeugt, welches bis zum Ende des Timeslots gültig ist. Die eingegebenen Attribute Benutzer und Passwort werden an das Sessioncookie zur weiteren Verwendung durch das SSH-Applet gebunden und das Sessioncookie an den Browser des Studenten geschickt, welcher dieses speichert. Danach präsentiert der Webserver dem Studenten die Laborseite des Kursmoduls.
- 7) In der Zwischenzeit wurde durch das Skript „update.pl“, welches einmal pro Minute gestartet wird, auf dem Internetportal das Passwort des aktuellen Benutzers aus der Moduldatenbank geholt und alle Systempasswörter in der Datei /etc/shadow, denen ein Laborgerät zugeordnet ist, auf dieses Passwort gesetzt. Auch diese Verbindung ist durch IPsec geschützt.
- 8) Klickt der Student im Webfrontend auf ein Symbol, dem ein Laborgerät zugeordnet ist, lädt sein Browser das SSH-Applet Mindterm auf seinen Rechner herunter und baut eine SSH-Verbindung zum Portal-Rechner auf.
- 9) Das Applet meldet sich automatisch am Portal an und benutzt als Benutzernamen den Namen des vom Studenten ausgewählten Gerätes (hier device1 oder device2) und als Passwort dasjenige, das der Student am Webfrontend des Kursmoduls angegeben hat und durch das Sessioncookie im Browser des Studenten gespeichert wurde.
- 10) Nach der erfolgreichen Anmeldung wird ein spezielles Login-Skript gestartet, welches den Studenten an das entsprechende Laborgerät weiterleitet. Diese Weiterleitung vom Portal zum Laborgerät wird über ein internes Netz übertragen und benutzt daher einen ungeschützten Anmeldemechanismus.
- 11) Der Student ist nun auf dem ausgewählten Laborgerät angemeldet und kann dieses benutzen.

6.3.2 Benutzerschnittstelle (Webfrontend)

Die Benutzerschnittstelle, welche das Bindeglied zwischen dem Benutzer und dem Portal darstellt, besteht aus in PHP geschriebenen Skripten, die vom Webserver des Portals ausgeführt und deren Ausgabe an den Browser des Benutzers geschickt werden. Das Webfrontend umfasst die folgenden vier Seiten:

- **Die Anmeldemaske (Abbildung 32)**
- **Die Fehlerseite (Abbildung 33)**
- **Die Laborseite des Kursmoduls (Abbildung 34)**
- **Die Abmeldeseite (Abbildung 35)**

Die Implementierung besteht aus vier PHP-Dateien, welche im Folgenden beschrieben werden:

- **ipsec.php**

Dies ist die Hauptdatei des Webfrontends und gibt je nach Zustand die Anmeldemaske, die Fehlerseite oder Abmeldeseite aus. Diese Datei benützt die unten beschriebene Datei `vitelsldap.inc.php` um zu kontrollieren, ob der Benutzer das Modul für die aktuelle Zeit wirklich reserviert hat. Ist dies der Fall, wird das oben beschriebene Sessioncookie erzeugt und an den Studentenbrowser gesendet. Des weiteren wird der Benutzer an die Datei `lab.php`, welche die Laborseite enthält, weitergeleitet.

- **lab.php**

Diese Datei zeigt die Laborseite mit den Geräten nur an, wenn vom Studentenbrowser ein gültiges Sessioncookie an den Webserver geschickt wird. Andernfalls wird eine leere Seite ausgegeben.

- **ipsec.inc.php**

Sie enthält Hilfsfunktionen, die HTML-Code kapseln.

- **vitelsldap.inc.php**

Diese Datei enthält die vorigen Kapitel bereits erwähnte VITELS-LDAP-Schnittstelle. Die darin enthaltene Funktion `get_slot_if_current_user` greift auf die LDAP-Daten des Reservierungssystems zu und liefert für die übergebenen Attribute Benutzer und Passwort die Start- und Endzeit des reservierten Timeslots.

6.3.3 Bildschirmabzüge der Benutzerschnittstelle und des SSH-Applets

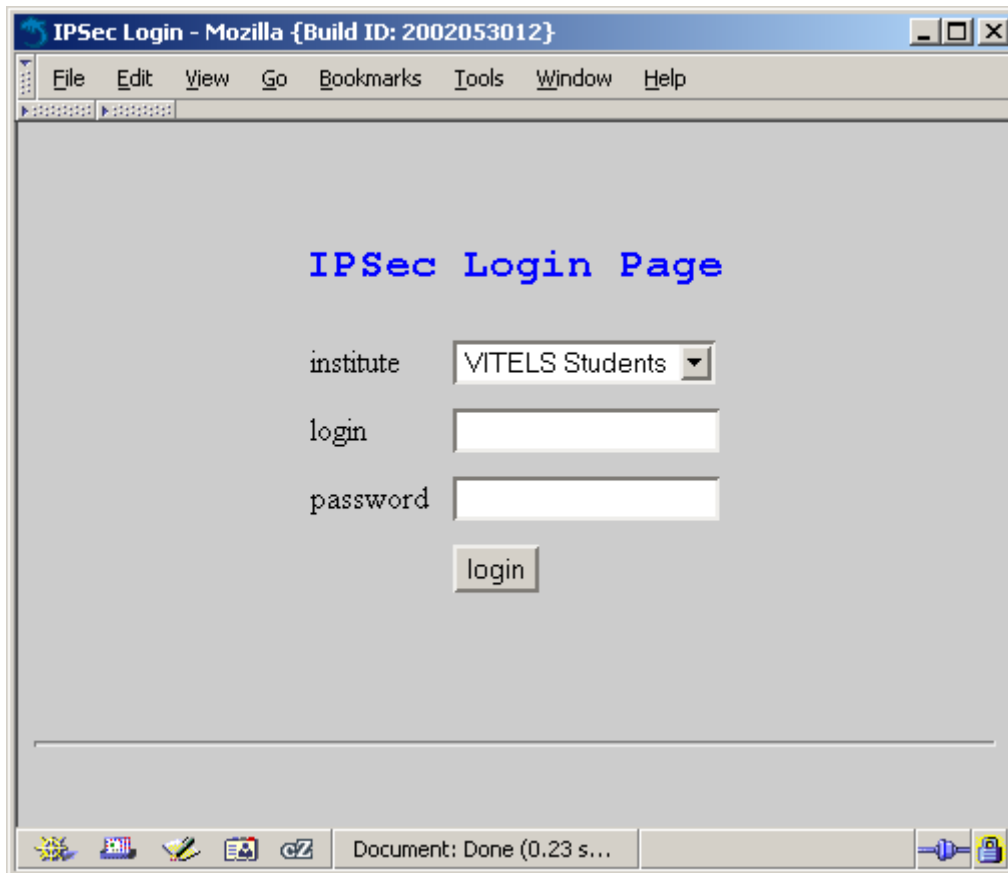


Abbildung 32: Anmeldemaske des IPsec-Moduls

Mittels der Anmeldemaske melden sich die StudentInnen am IPsec-Modul an. Sie geben hier ihre persönliche eindeutige Benutzerkennung und ihr Passwort ein. Diese Information wird mit den Daten im Reservierungssystem verglichen und entschieden, ob der Benutzer das Modul für die aktuelle Zeit reserviert hat oder nicht.

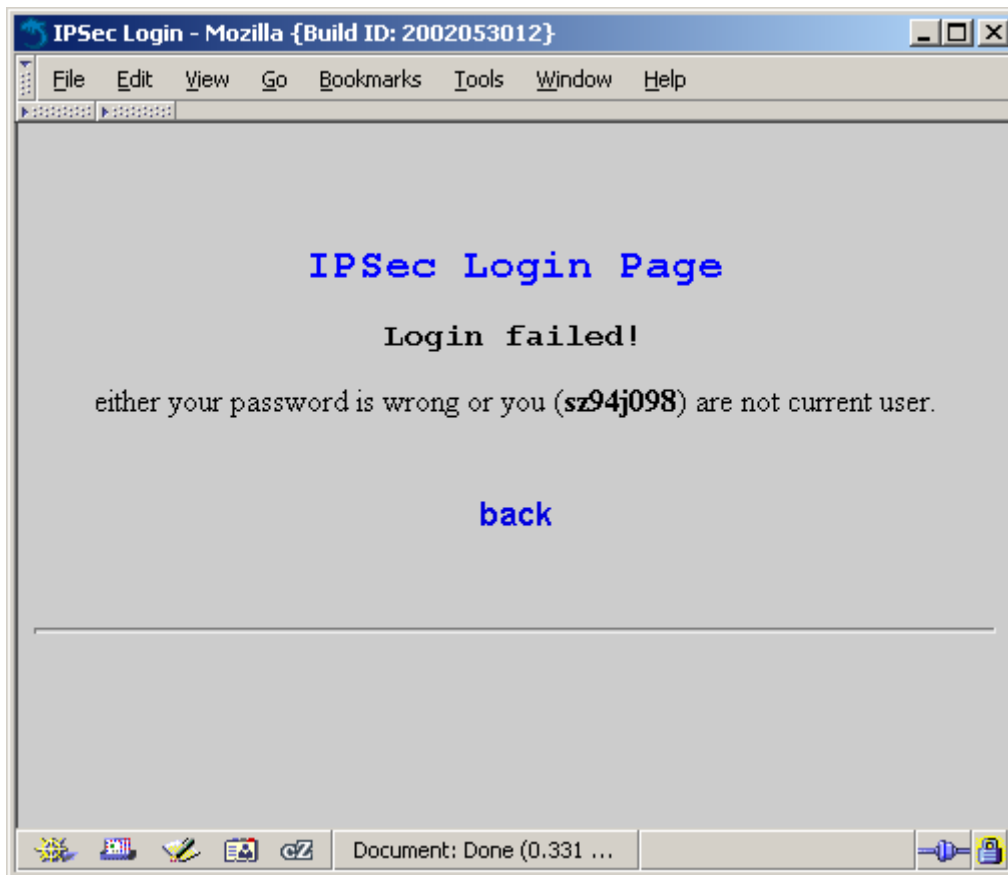


Abbildung 33: Fehlerseite des IPsec-Moduls

Hat der Benutzer das Passwort falsch eingegeben oder das Modul zum aktuellen Zeitpunkt nicht reserviert, präsentiert ihm das Frontend diese Fehlerseite.

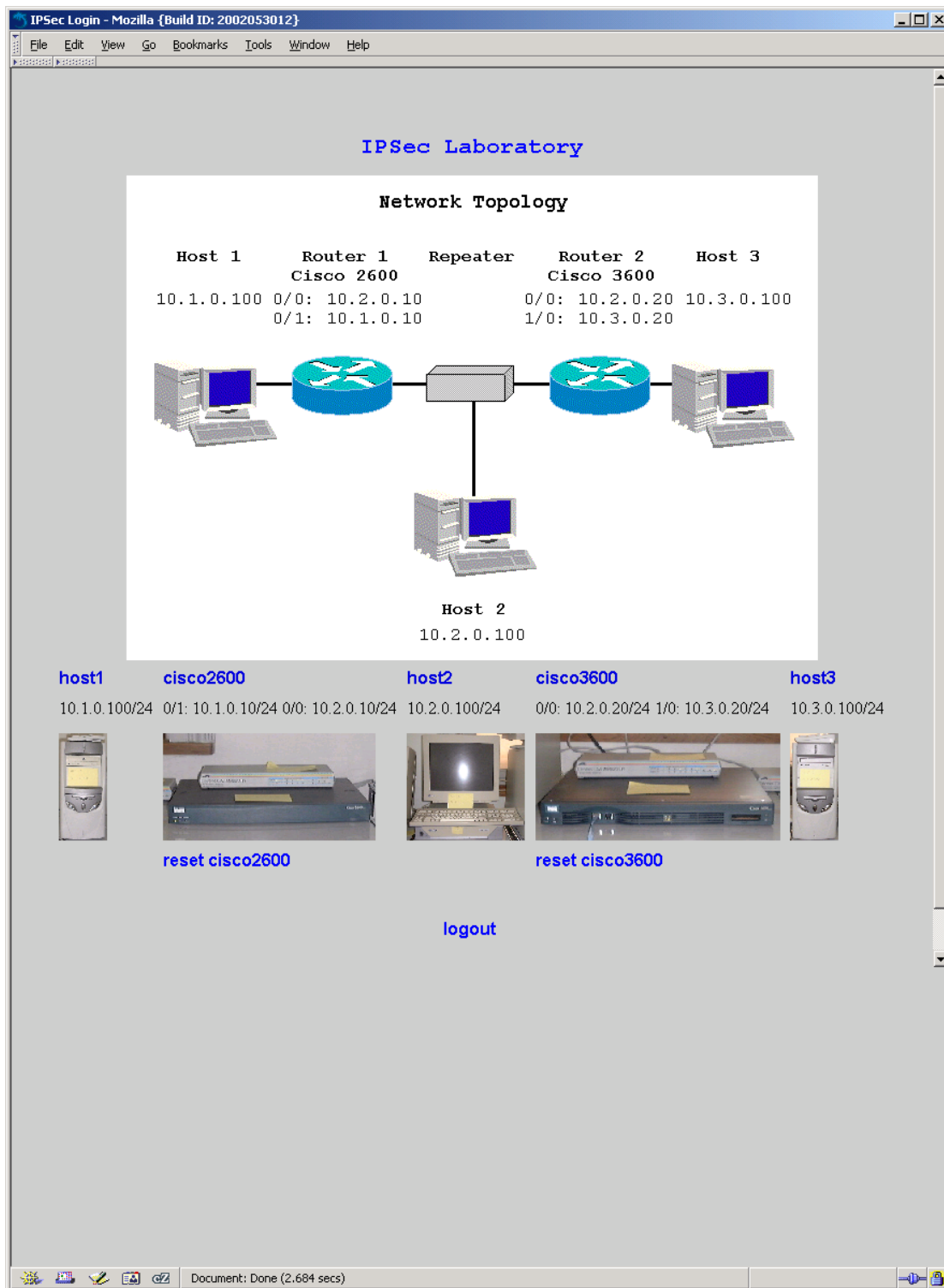


Abbildung 34: Hauptseite (Labor) des IPsec-Moduls

Dies ist die Hauptseite des IPsec-Moduls und wird nach erfolgreichem Anmelden demjenigen Benutzer präsentiert, der für die aktuelle Zeit das Modul reserviert hat. Durch Mausklick auf ein Host- oder Routersymbol wird das SSH-Applet heruntergeladen und gestartet. Dieses baut eine sichere Verbindung zum Internetportal auf und ermöglicht dem Benutzer den exklusiven Zugriff auf das ausgewählte Gerät (siehe auch Abbildung 36 und Abbildung 37). Die Links „reset cisco2600“ und „reset cisco3600“ ermöglichen es dem Benutzer, die Router in einen

definierten Ausgangszustand zurückzusetzen (siehe auch Kapitel 7.2). Mittels Klick auf „logout“ kann sich der Benutzer abmelden.

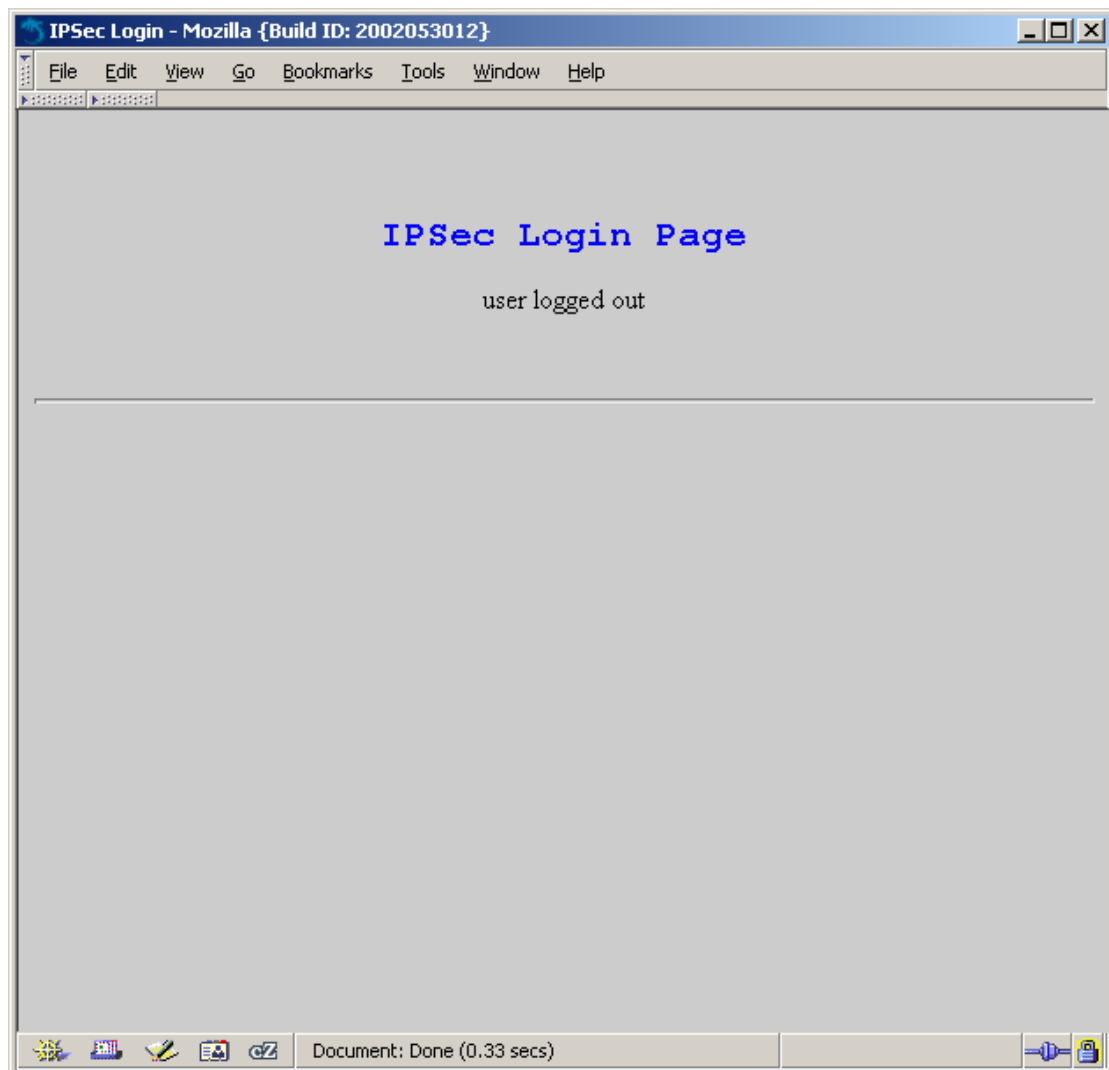


Abbildung 35: Abmeldeseite des IPsec-Moduls

Nach erfolgreicher Abmeldung wird obenstehende Abmeldeseite dargestellt.

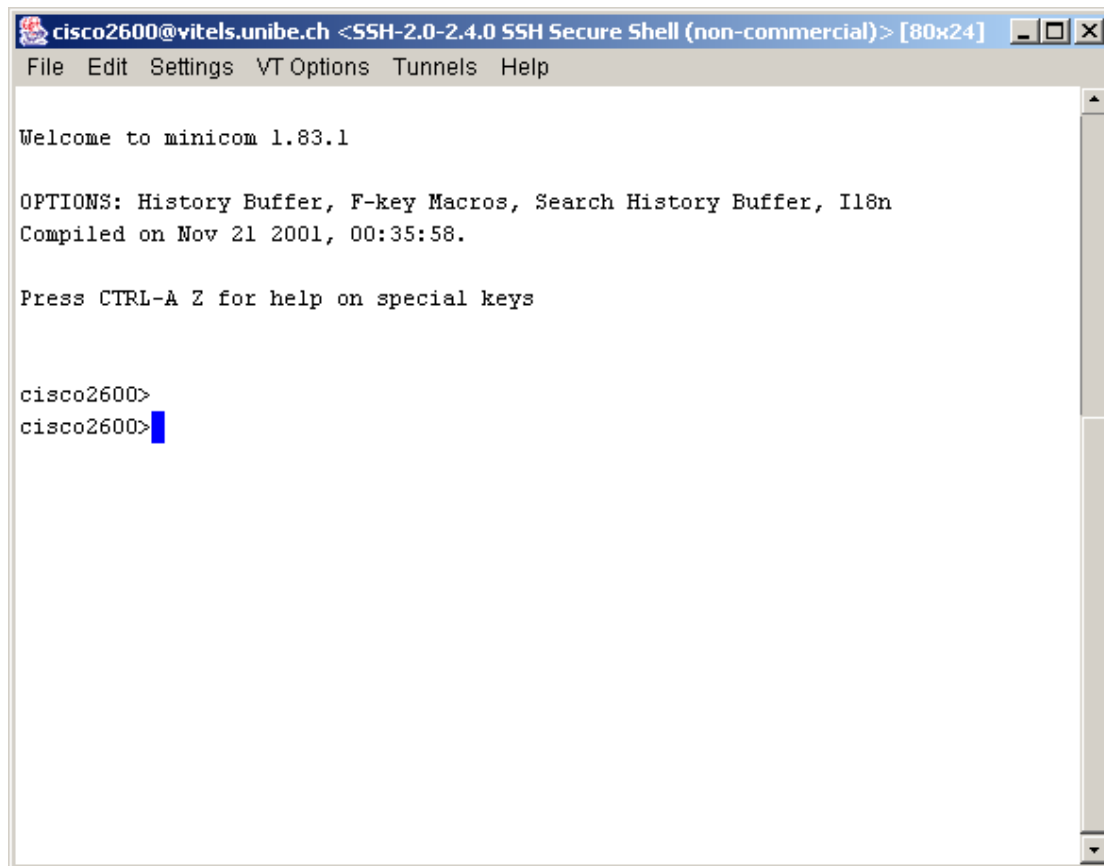
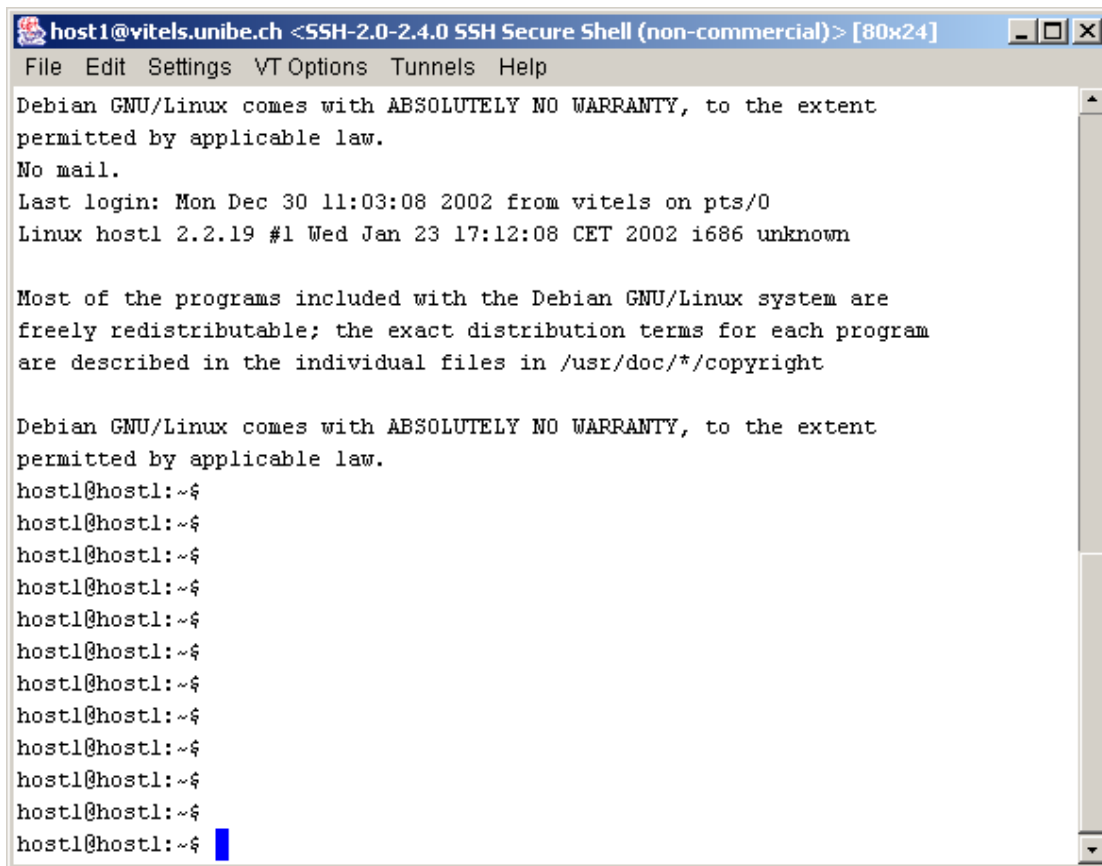


Abbildung 36: Zugriff auf den Router cisco2600 mittels SSH-Applet

Abbildung 36 zeigt das durch Klick auf den Router auf den Studentenrechner heruntergeladene SSH-Applet, welches via Portal den Zugriff auf den Cisco Router 2600 ermöglicht. Nach der Anmeldung am Internetportal wurde automatisch die Applikation Minicom gestartet (siehe auch Kapitel 7.1.1).



```
host1@vitels.unibe.ch <SSH-2.0-2.4.0 SSH Secure Shell (non-commercial)> [80x24]
File Edit Settings VT Options Tunnels Help
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No mail.
Last login: Mon Dec 30 11:03:08 2002 from vitels on pts/0
Linux host1 2.2.19 #1 Wed Jan 23 17:12:08 CET 2002 i686 unknown

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
host1@host1:~$
```

Abbildung 37: Zugriff auf den Rechner host1 mittels SSH-Applet

Diese Abbildung zeigt das SSH-Applet nach Klick auf das Symbol des Rechners host1. Der Benutzer ist auf diesem Rechner angemeldet und hat dadurch Zugriff auf diesen Rechner. Dies ermöglicht es ihm, Messungen durchzuführen.

7 Implementierung des Internetportals für das Fernkursmodul IPSec

In diesem Kapitel wird auf die Anpassungen eingegangen, um die allgemeine Realisierung des Internetportals für das Fernkursmodul IPSec zu implementieren. Das VITELS-Fernkursmodul IPSec basiert auf dem traditionellen Praktikumsmodul IPSec, welches im Kapitel 4 beschrieben ist. Die folgende Abbildung zeigt das implementierte Internetportal und die dazugehörigen Laborgeräte.

Das Portal besitzt drei interne Netzwerkschnittstellen mittels welchen der Zugriff auf alle drei Teilnetze möglich ist. Mittels der vierten Netzwerkschnittstelle ist es mit dem Internet verbunden und hat Zugriff auf das Reservationssystem. Zusätzlich führen zwei Leitungen von den seriellen Schnittstellen zu den Con-Ports der Router, durch welche der immerwährende Zugriff des Portals auf die Router sichergestellt wird, da die Con-Ports (im Gegensatz zu den Netzwerkschnittstellen) nicht abgestellt werden können. Die dritte serielle Leitung führt zur Relaiskarte, welche die Stromversorgung der Router steuert. Dadurch ist es dem Portal möglich, die Router beliebig ein- und auszuschalten.

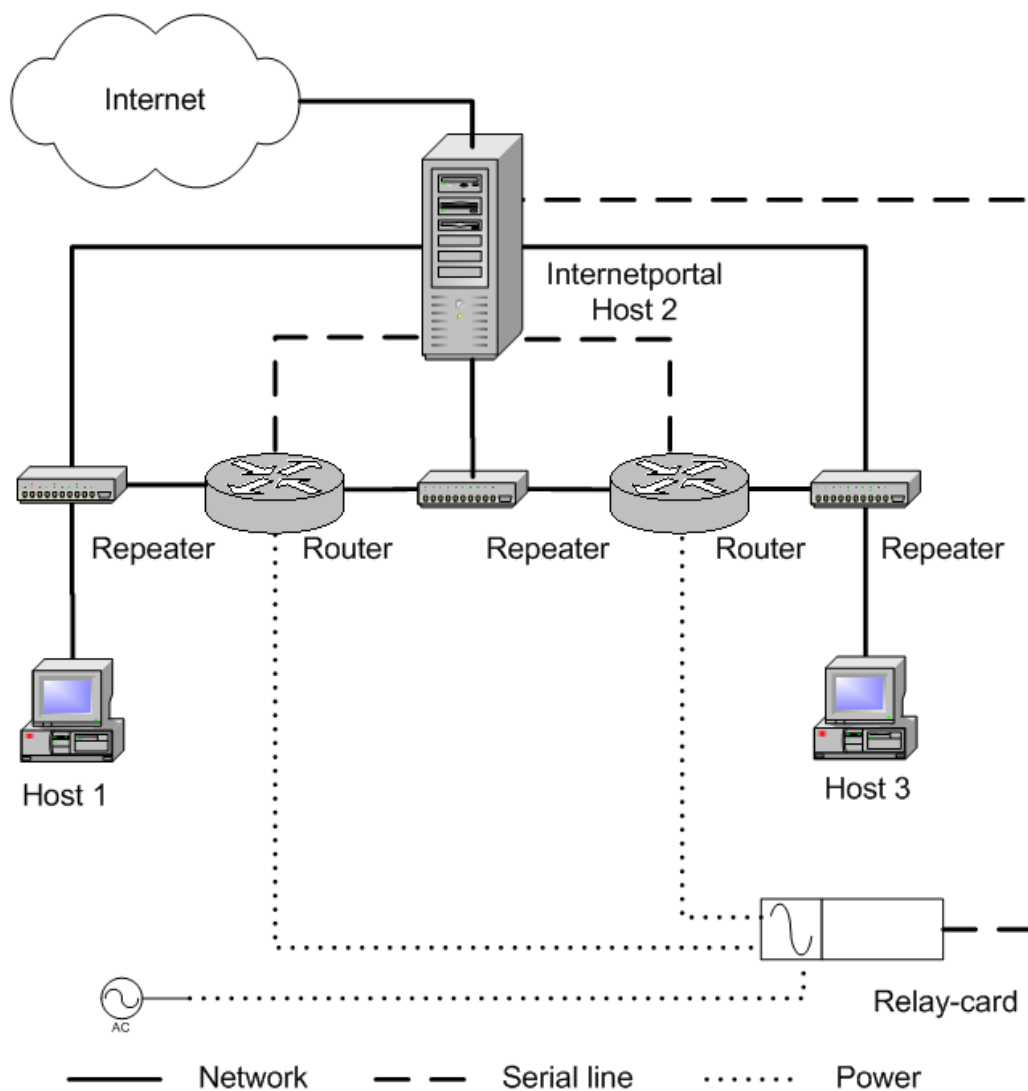


Abbildung 38: Internetportal und Laborgeräte des VITELS-Fernkursmoduls IPSec

7.1 Unterschiede zwischen der allgemeinen Realisierung des Internetportals und der Implementierung für das Fernkursmodul IPSec

7.1.1 Anmeldeweiterleitung an Linux-Rechner bzw. Cisco Router

Wie im Kapitel 6.3 beschrieben, meldet sich der Benutzer des Fernkursmoduls am Internetportal an. Aufgrund des dem Portal übermittelten Anmeldenamens (login), entscheidet das Portal, an welches Laborgerät der Benutzer weitergeleitet wird. Die folgende Abbildung 39 zeigt dies für die Laborgeräte host1 (ein Linux-Rechner) und cisco2600 (ein Cisco Router):

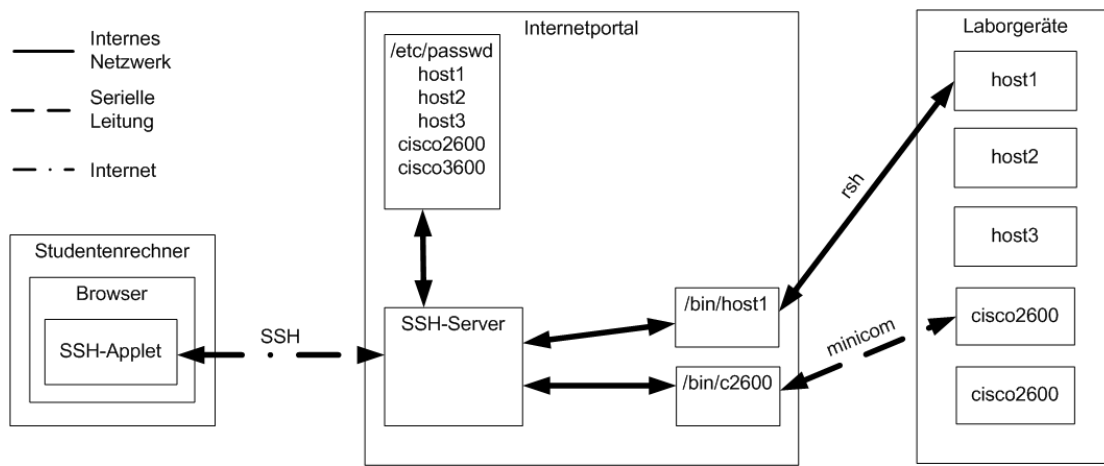


Abbildung 39: Anmeldeweiterleitung an Linux-Rechner und Cisco Router

Meldet sich ein Benutzer am Portal mit dem Namen „host1 an, wird das folgende Anmeldeskript ausgeführt. Dieses baut mittels des Befehls `rsh` eine Remote-Shell-Verbindung [68] zum Rechner `host1` auf. Auf diesem Rechner läuft der entsprechende Serverprozess `rshd`.

```
#!/bin/bash
# /bin/host1
/usr/bin/rsh host1
exit
```

Datei 1: Loginskript des Portalusers `host1`

Damit sich der Benutzer am Linux-Rechner `host1` nicht noch einmal anmelden muss, wurde die folgende `rhost`-Datei erstellt. Sie erlaubt, dass sich der User `host1` auf dem Rechner `host1` über das interne Ethernet-Netzwerk ohne Passwort anmelden darf, sofern er vom Rechner `vitels` (dem Portal) kommt.

```
# host1:/home/host1/.rhosts
vitels host1
```

Datei 2: `rhost` Konfiguration des Users `host1` auf dem Linux-Rechner `host1`

Im Gegensatz zu den Linux-Rechnern werden die Router nicht übers Netzwerk konfiguriert, sondern mittels des auf dem Portal gestarteten Terminalprogramms Minicom. Dieses wird durch das Loginskript des Portalusers cisco2600 gestartet. Es ist vorkonfiguriert und baut über die erste serielle Schnittstelle die Verbindung zum Router auf.

```
#!/bin/bash
/usr/bin/minicom -C /home/cisco2600/minicom.log ttyS0
exit
```

Datei 3: Loginskript des Portalusers cisco2600

7.1.2 Integration des Rechners host2 in das Portal

Eine Besonderheit ist der host2, denn er ist mittels einer sogenannten Change-Root-Umgebung [69] im Internetportal integriert. Change-Root-Umgebungen sind empfehlenswert, wenn man einem Benutzer den Zugriff auf einen ganz bestimmten Teil (-baum) des baumartig aufgebauten Dateisystems beschränken will. Dies wird dadurch erreicht, dass die virtuelle Dateisystemwurzel (root) dieses Benutzers in ein beliebiges Unterverzeichnis des realen Dateisystems verschoben wird. Die Abbildung 40 illustriert die beschriebene Hierarchie:

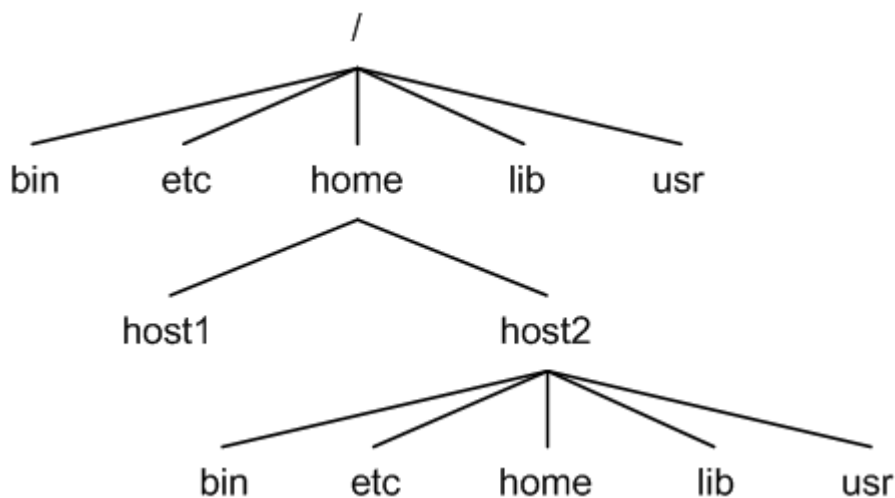


Abbildung 40: Dateisystemstruktur der Change-Root-Umgebung

Meldet sich der Benutzer host2 am Portal an, ist sein Wurzelverzeichnis nicht „/“, sondern das Verzeichnis „/home/host2“. Dies hat die Konsequenz, dass der Benutzer host2 keine Möglichkeit hat, aus diesem virtuellen Wurzelverzeichnis in höhere Verzeichnisebenen zu wechseln. Anders gesagt ist alles, was oberhalb seines Wurzelverzeichnisses liegt, für ihn unsichtbar, was bedeutet, dass alle Befehle, Bibliotheken und Konfigurationsdateien, die er benötigt, aus den Verzeichnissen des realen Dateisystems in die entsprechenden Verzeichnisse des virtuellen Wurzelverzeichnisses kopiert werden müssen. Daraus folgt, dass einem solchen Benutzer nur diejenigen Befehle zur Verfügung gestellt werden können, die er für

seine Arbeit benötigt. Im Falle des IPSec-Moduls werden dem Benutzer host2 die Befehle ping, traceroute (Kapitel 4.6), Tcpdump (Kapitel 4.8) und der Editor vi zur Verfügung gestellt.

Im folgenden wird der Ablauf der Anmeldung des Benutzers host2 beschrieben und die dafür notwendigen Konfigurationen erläutert. Der Ablauf besteht aus drei Schritten, wie die Abbildung 41 zeigt:

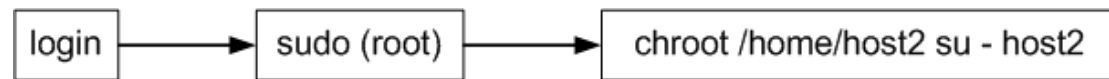


Abbildung 41: Anmeldeablauf des Chroot-Root-Benutzers host2

In einem ersten Schritt wird der Benutzer authentifiziert und bei Erfolg dessen Shell gestartet, die in der Datei /etc/passwd (Datei 4) an letzter Stelle eingetragen ist. Im Falle des Benutzers host2 wird also das Skript /bin/chroot-shell (Datei 5) gestartet, welches die beiden Befehle sudo und chroot ausführt. Diesem Skript wird über die Variable \$USER der Benutzername, hier also host2 übergeben und es unterscheidet, ob es mit oder ohne zusätzliche Parameter aufgerufen wird.

```
host1:x:1005:1005:IP Security Lab User,,,:/home/host1:/bin/host1
host2:x:1003:1003:IP Security Lab User,,,:/tmp:/bin/chroot-shell
host3:x:1006:1006:IP Security Lab User,,,:/home/host3:/bin/host3
```

Datei 4: /etc/passwd-Einträge der Portalbenutzer host1, host2 und host3

```
#!/bin/bash

if [ "$1" = "-c" ]; then
    i=0;
    PARAMS="";
    for param in $*; do
        if [ $i -gt 0 ]; then
            PARAMS="$PARAMS $params";
        fi
        let i++;
    done;
    sudo /usr/sbin/chroot /home/$USER /bin/su - \
    $USER -c "$PARAMS"
else
    sudo /usr/sbin/chroot /home/$USER /bin/su - $USER
fi;
```

Datei 5: Shellskript /bin/chroot-shell des Benutzers host2

Im zweiten Schritt wird mittels des Befehls sudo der nachfolgende Befehl chroot als Administrator aufgerufen, da chroot nicht als normaler Benutzer aufgerufen werden kann. Damit der Benutzer host2 diesen sudo-Befehl ausführen darf, muss ihm dies explizit in der Datei /etc/sudoers erlaubt werden. Durch den Eintrag (zweite Zeile in Datei 6) wird definiert, dass der Benutzer host2 als Administrator auf der Maschine vitels (dem Portal) den Befehl (und nur diesen) /usr/sbin/chroot /home/host2 /bin/su - host2 ausführen darf, ohne ein Passwort angeben zu müssen.

```
# chroot jail for user host2 in /home/host2
host2 vitels=NOPASSWD: /usr/sbin/chroot /home/host2 /bin/su - host2
```

Datei 6: sudo-Konfigurationsdatei /etc/sudoers

Im dritten und letzten Schritt wird mittels chroot die Verschiebung der Dateisystemwurzel vollzogen. Chroot erwartet zwei Parameter. Der erste gibt das neue Wurzelverzeichnis an, während der zweite den Befehl definiert, der im neuen Wurzelverzeichnis ausgeführt werden soll. Durch

```
/usr/sbin/chroot /home/host2 /bin/su - host2
```

wird als neues Wurzelverzeichnis /home/host2 definiert und mittels des Befehls su wieder von der Administratorebene zur Benutzerebene host2 gewechselt. Nun ist der Benutzer host2 am Portal angemeldet und sein virtuelles Wurzelverzeichnis befindet sich im Verzeichnis /home/host2.

7.2 Automatisches Löschen eines gesetzten Cisco Router Administrationspasswords

Wie schon in Kapitel 4.5 erwähnt, erfordert die Konfiguration eines Cisco Routers Administrationsrechte. Der Zugang zum Administrationsmodus ist normalerweise durch ein Passwort geschützt. Beim Fernkursmodul IPsec wird jedoch darauf verzichtet, da sich die Benutzer schon vorher authentifizieren müssen. Trotzdem besteht die Möglichkeit, dass die StudentInnen - aus welchen Gründen auch immer – ein solches Passwort setzen und damit den oder die Router für den nächsten Benutzer sperren. Aus diesem Grund musste ein Mechanismus entwickelt und implementiert werden, der es erlaubt ein solches Passwort automatisch wieder zu löschen und die Konfiguration zurückzusetzen. In den folgenden beiden Abschnitten werden die beide Ansätze beschrieben.

7.2.1 Ansatz 1: Der Serial-Line Sniffer

Der Serial-Line Sniffer (SLSNIF, [70]) ist eine Applikation, die auf dem Portal gestartet wird, und sich zwischen das Programm Minicom und die serielle Schnittstelle (an die der Router angeschlossen ist) stellt. Nachdem diese Applikation gestartet wurde, greift sie auf die reale serielle Schnittstelle zu und präsentiert Minicom eine sogenannte Pseudo Serielle Schnittstelle. Der Sniffer macht nichts anderes, als die Eingaben von Minicom entgegen zu nehmen und an den Router weiterzureichen. Antworten des Routers werden gelesen und an Minicom zurückgegeben. Zusätzlich wird alles, was von der Applikation in beiden Richtungen gelesen wird, in eine Logdatei geschrieben.

Falls nun ein Benutzer des Fernkursmoduls auf einem Router ein Administrationspassword setzt, wird dieses vom Sniffer aufgezeichnet und in die Logdatei geschrieben. Aus dieser kann es nachträglich extrahiert werden und damit der Administrationsmodus wieder zugänglich gemacht werden.

Dieser Ansatz wurde darum verworfen, da sich StudentInnen auf einem der beiden Linux-Rechner host1 und host3 anmelden und von dort eine Telnet-Verbindung auf die Router aufbauen können. Setzen sie nun via diese Verbindung ein Administrationspassword auf dem Router, wird dies vom Serial-Line Sniffer nicht mitgelesen und somit nicht in die Logdatei geschrieben, was die Sperrung des Router für den nächsten Benutzer zur Folge hätte

7.2.2 Ansatz 2: Die Cisco Password Recovery Solution

Der zweite Ansatz basiert auf der von Cisco dokumentierten Password Recovery Solution [71] mittels welcher ein gesetztes Administrationspasswort gelöscht werden kann. Um diese durchführen zu können muss normalerweise physikalischer Zugang zum Router vorhanden sein.

7.2.2.1 Manuelles Vorgehen

Die Password-Recovery Solution wird mittels den folgenden Schritten durchgeführt:

- **Anschliessen der seriellen Schnittstelle eines Rechners an den Con-Port des Routers**
- **Starten der Terminalapplikation auf dem Rechner**
- **Auswalten des Routers**
- **Einschalten des Routers**
- **Senden der Breaksequenz**

Die Breaksequenz ist eine schnelle Folge von 0 und 1 Bits, die über die serielle Schnittstelle an den Router geschickt wird. Sie dient normalerweise dazu, einen Befehl abubrechen. Die meisten Terminalapplikation definieren hierzu eine bestimmte Taste oder Tastenkombination.

Wird die Breaksequenz innerhalb 30 Sekunden nach dem Einschalten des Routers gesendet, wechselt er in den sogenannten Rom-Mon, den Rom-Monitor. In diesem speziellen Zustand ist das Betriebssystem des Routers (Internet Operating System, IOS) noch nicht geladen und ein Administrator kann von einem speziellen Server, einem Trivial File Transfer Protocol (TFTP) [72] -Server ein anderes oder neues IOS auf den Router laden.
- **Umstellen des Konfigurationsregisters**

Im Rom-Mon ist es nun möglich das Konfigurationsregister, welches die Einsprungsadresse zum IOS enthält, so zu verstellen, dass der Router ohne Konfiguration, gestartet wird. Dies bedeutet einerseits, dass alle Ethernet-Netzwerkschnittstellen unkonfiguriert sind, das Routing abgeschaltet ist und die VPN-Konfiguration gelöscht ist. Andererseits ist in diesem Zustand ein gesetztes Administrations-Passwort des Routers gelöscht.
- **Neustarten des Routers**

Anschliessend wird der Router neugestartet.
- **Zurückstellen des Konfigurationsregisters**

Nachdem der Router neugestartet ist, ist es möglich, ohne ein Passwort anzugeben, in den Administrationsmodus zu wechseln und das Konfigurationsregister wieder auf den ursprünglichen Wert zurückzusetzen.

- **Neustarten des Routers**

Nach einem zweiten Neustart ist der Router wieder verfügbar.

Die hierzu nötigen Befehle sind im folgenden festgehalten.

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE
(fc1)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
PC = 0xffff0a530, Vector = 0x500, SP = 0x680127b0
C2600 platform with 32768 Kbytes of main memory

PC = 0xffff0a530, Vector = 0x500, SP = 0x8000488c
```

Abbildung 42: Routermeldung nach Einschalten

```
monitor: command "boot" aborted due to user interrupt
rommon 1 >
```

Abbildung 43: Routermeldung nach Break-Sequenz

```
rommon 1 > confreg 0x2142

You must reset or power cycle for new config to take
effect
rommon 2 >
```

Abbildung 44: Umstellen des Konfigurationsregisters

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE
(fc1)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
C2600 platform with 32768 Kbytes of main memory

program load complete, entry point: 0x80008000, size:
0x7fb550

cisco 2621 (MPC860) processor (revision 0x100) with
29696K/3072K bytes of memor.
Processor board ID JAB0318062X (3013597390)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK203S-M), Version
12.1(5), RELEASE SOFTWARE (fc)
```

```
Copyright (c) 1986-2000 by cisco Systems, Inc.  
Compiled Wed 25-Oct-00 11:02 by cmong  
00:01:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface  
FastEthernet0/0, chann  
Router>  
Router>
```

Abbildung 45: Routermeldung nach Neustart

```
Router>enable  
Router#configure  
Configuring from terminal, memory, or network [terminal]?  
Enter configuration commands, one per line. End with  
CNTL/Z.  
Router(config)#config-register 0x2102  
Router(config)#exit  
Router#  
00:03:31: %SYS-5-CONFIG_I: Configured from console by  
console  
Router#  
  
Router#copy running-config startup-config  
Destination filename [startup-config]?  
Building configuration...  
[OK]  
Router#
```

Abbildung 46: Zurückstellen des Konfigurationsregisters

```
Router#reload  
Proceed with reload? [confirm]  
  
00:06:19: %SYS-5-RELOAD: Reload requested  
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE  
(fc1)  
Copyright (c) 1999 by cisco Systems, Inc.  
TAC:Home:SW:IOS:Specials for info  
C2600 platform with 32768 Kbytes of main memory  
  
program load complete, entry point: 0x80008000, size:  
0x7fb550  
  
cisco 2621 (MPC860) processor (revision 0x100) with  
29696K/3072K bytes of memor.  
Processor board ID JAB0318062X (3013597390)  
M860 processor: part number 0, mask 49  
Bridging software.  
X.25 software, Version 3.0.0.  
2 FastEthernet/IEEE 802.3 interface(s)
```

```

32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK203S-M), Version
12.1(5), RELEASE SOFTWARE (fc)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 25-Oct-00 11:02 by cmong
Router>

```

Abbildung 47: Routermeldung nach zweitem Neustart

7.2.2.2 Die Relaiskarte

Um die im vorigen Kapitel 7.2.2.1 beschriebene Prozedur zu automatisieren und vom Portal aus zu starten, wurde eine Relaiskarte [73] der Firma Conrad zwischen den Stromanschluss des Routers und der Steckdose geschaltet, wie die Abbildung 38 illustriert. Diese Relaiskarte besitzt acht Relais, von denen für unsere Zwecke jedoch nur zwei (je einen pro Router) benutzt werden. Zusätzlich befindet sich auf der Platine ein Mikrokontroller, der einerseits die Relais kontrolliert und andererseits über die serielle Schnittstelle mit einem Rechner, in unserem Fall dem Internetportal kommuniziert. Die Kommunikation basiert auf einem proprietären Protokoll [74], welches 4-Byte-Rahmen Kommandos benutzt. Für jeden Kommandorahmen der an den Mikrokontroller geschickt wird, wird ein Antwortrahmen an das Portal zurückgeschickt. Bis zu acht solcher Karten können kaskadiert werden. Der Kontroller kennt die folgenden Kommandos:

- **No Operation**
Führt keine Aktion aus und kann zu Prüfzwecken verwendet werden.
- **Setup**
Initialisiert die Relaiskarte und ordnet ihr eine eindeutige Adresse zu. Gibt die Versionsnummer der Mikrokontroller-Software zurück.
- **Get Port**
Gibt den Wert zurück, dessen Bitmuster dem Zustand der acht Relais entspricht.
- **Set Port**
Setzt die acht Relais auf einen anzugebenden Wert, dessen Bitmuster dem Zustand der acht Relais entspricht.

Die nachfolgende Tabelle zeigt, wie diese Befehle und deren Antworten in die 4-Byte-Rahmen kodiert werden:

Nummer	Kommando	Kommandorahmen				Antwortrahmen			
		0	Adresse	x	XOR	255	Adresse	x	XOR
0	NoOperation	0	Adresse	x	XOR	255	Adresse	x	XOR
1	Setup	1	Adresse	x	XOR	254	Adresse	Info	XOR
2	Get Port	2	Adresse	x	XOR	253	Adresse	Daten	XOR
3	Set Port	3	Adresse	Daten	XOR	253	Adresse	x	XOR

Tabelle 7: Kodierung der Befehls- und Antwortrahmen

7.2.2.3 Der implementierte Relaiskartentreiber

Um die Relaiskarte vom Portal aus ansprechen zu können, musste ein Treiber geschrieben werden, der das proprietäre Protokoll zwischen dem Mikrokontroller und dem Steuerungsrechner (dem Portals) implementiert. Dieser Treiber wurde als Bibliothek in der Sprache PERL geschrieben und ist von der Bibliothek `Device::SerialPort` abhängig. Diese freie Bibliothek bietet Funktionen für die Kommunikation mit Geräten, die über serielle Schnittstellen angebunden werden.

Der implementierte Relaiskartentreiber `RelaisLib.pm` stellt die folgenden Funktionen zur Verfügung:

- **relaiscard_tty_open**

Diese Funktion öffnet die übergebene serielle Schnittstelle (`/dev/ttySx`) und konfiguriert deren Übertragungseigenschaften wie die Anzahl der Datenbits (8), die Baudrate (19200), die Parität (keine) und die Anzahl der Stoppbits (1). Der Rückgabewert ist ein Handle auf die geöffnete Schnittstelle.

- **relaiscard_tty_close**

Diese Funktion schliesst das ihr übergebene Schnittstellenhandle, welches man von der Funktion `relaiscard_tty_open` erhalten hat.

- **encode_frame**

Dieser Funktion übergibt man die drei ersten Zahlen aus der Spalte Kommandorahmen der Tabelle 7. Daraus wird mittels der XOR-Funktion die Prüfsumme (die vierte Zahl) berechnet. Diese vier Zahlen werden danach in einen 4-Byte-String konvertiert, welcher nun an den Relaiskartenkontroller geschickt werden kann.

Die Funktion liefert als Rückgabewert den kodierten 4-Byte-Rahmen

- **decode_frame**

Diese Funktion stellt das Gegenstück der vorigen Funktion dar und dekodiert einen 4-Byte-Rahmen zurück in die darin enthaltenen vier Zahlen und gibt diese zurück.

- **check_frame**

Als Eingabewert wird ein kodierter 4-Byte-Rahmen erwartet. Die Funktion überprüft, ob die Prüfsumme des Rahmen richtig ist, also ob $\text{Byte1 XOR Byte2 XOR Byte3} = \text{Byte4}$ ist. Ist dem so, wird wahr ansonsten falsch zurückgegeben.

- **relaiscard_send_command**

Dieser Funktion übergibt man das Handle auf die geöffnete Schnittstelle, die Adresse der Relaiskarte, das Relaiskartenkommando (Setup, Get Port) und im Falle des Kommandos Set Port den Wert auf den die Relais gesetzt werden sollen. Das Kommando wird mittels von `encode_frame` kodiert, an den Relaiskontroller geschickt, dessen Antwortrahmen dekodiert und als Rückgabewert zurückgegeben.

- **get_relais_status**

Dieser Funktion übergibt man das Handle der geöffneten Schnittstelle und sie liefert unter Nutzung der Funktion `relaiscard_send_command` den Zustand der Relais als Zahl zurück (0: Alle Relais aus, 255: Alle Relais ein)

Der vollständige Quellcode des Relaiskartentreibers befindet sich im Anhang in der Datei 7.

7.2.2.4 Die implementierte Routerbibliothek

Das Reset-Skript, welches die in Kapitel 7.2.2 beschriebene Password Recovery Solution automatisiert, nutzt einerseits den Relaiskartentreiber, um damit die Stromversorgung der Router zu unterbrechen und andererseits eine weitere in PERL implementierte Bibliothek `RouterLib.pm` (siehe auch Datei 8), die folgende Funktionen zur Verfügung stellt:

- **create_lockfile**

Wird ein Router zurückgesetzt, dauert dies ungefähr fünf Minuten. Damit während dieser Zeitspanne nicht ein zweiter Reset gestartet werden kann, erzeugt diese Funktion für den entsprechenden Router eine sogenannte Lock-Datei. Existiert diese Datei und ist sie jünger als sechs Minuten, wird kein Reset durchgeführt.

- **delete_lockfile**

Löscht die obengenannte Lock-Datei.

- **exist_lockfile**

Prüft, ob eine Lock-Datei existiert.

- **is_lockfile_new**

Prüft, ob die Lock-Datei älter als die maximale Dauer eines Routerresets ist.

- **send_router_command**

Diese Funktion schickt dem Router einen Befehl über die serielle Leitung und wartet darauf, dass der Router diesen abgearbeitet hat. Ein Befehl wird dem Router geschickt, in dem die Zeichenfolge des Befehls direkt in die serielle Schnittstelle geschrieben (mittels des Befehls `write` der Bibliothek `Device::SerialPort`) wird.

- **reset_router**

Dieser Funktion übergibt man den Namen des zu zurücksetzenden Routers. Mittels der Relaiskartenbibliothek schaltet sie den entsprechenden Router aus und wieder ein. Danach werden via die Funktion `send_router_command` die Befehle der Password Recovery Solution Schritt für Schritt abgearbeitet und somit ein gesetztes Passwort gelöscht.

7.2.2.5 Das implementierte Reset-Skript

Auf der Grundlage der beiden beschriebenen Bibliotheken `RelaisLib.pm` und `RouterLib.pm` wurde ein Skript in PERL implementiert, welches den Zustand der Lock-Datei kontrolliert und den einen oder beide Router zurücksetzt. Des weitern schreibt es entsprechende Meldungen über seinen Zustand in das Logbuch des Internetportals. Der Ablauf dieses Skriptes `pw_reset.pl` wird im folgenden skizziert. Der Quellcode befindet sich im Anhang in der Datei 9.

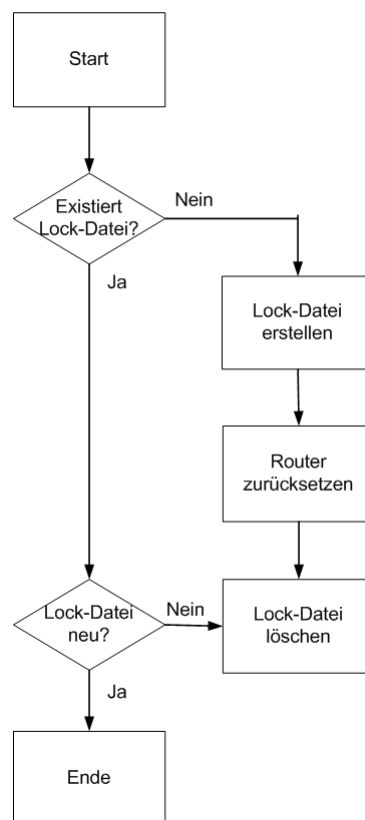


Abbildung 48: Ablauf des Skriptes `pw_reset.pl`

8 Zusammenfassung und Ausblick

8.1 Resultate

Im Rahmen dieser Arbeit wurde ein Konzept entworfen, um reale Laborgeräte wie Cisco Router und Linux-Rechner mittels eines Internetportals für den Fernzugriff anzubieten. Im weitesten Sinne kann irgendein Gerät, das eine Schnittstelle zu einem Standard-Rechner besitzt, an das Internetportal angeschlossen werden. Dabei wurde der Augenmerk besonders auf die Sicherheit der übertragenen persönlichen Daten der StudentInnen gelegt. Ein weiterer wichtiger Punkt war die Robustheit des Systems, damit die Administratoren nicht viel Zeit dafür aufwenden müssen, um das System am Laufen zu halten. Zusätzlich sollte die Echtheit der Konfiguration von realen Geräten durch den Fernzugriff via Webbrowser nicht verloren gehen, was durch den Zugriff mittels einer Shell erreicht werden konnte. Aufgrund der relativ schnellen Implementation des Prototyps und dessen erstes Anbieten an die StudentInnen konnten wichtige Erfahrungen gemacht werden. Als Beispiel sei hier der erste Ansatz zum Löschen des Routeradministrationspasswortes genannt, der nach dem Betatest als ungenügend verworfen werden musste. Der daraus gefolgerte, zweite etwas aufwändigere Ansatz zeigte jedoch, dass es doch möglich ist, den StudentInnen vollen Administrationszugriff auf die Router anzubieten und ein gesetztes Passwort wieder zu entfernen.

8.2 Diskussion

Nach mittlerweile drei produktiven Tests des VITLES-Fernkursmoduls IPSecurity mit mehreren Studentinnen und Studenten, musste festgestellt werden, dass der Übergang von einem traditionellen Labor zu einem Fernkurslabor einige Schwierigkeiten aufwirft. Zum ersten sei hier der Vorteil des Lernens in Gruppen genannt. Während im traditionellen Labor die StudentInnen in Gruppen die Arbeit verrichten und somit die Möglichkeit haben, sich auszutauschen, besteht beim Fernkurs das Problem, dass sie die Arbeiten alleine erledigen müssen und auf sich selbst gestellt sind. Des weiteren ist es bei einem Fernkurs nicht möglich, schnell in das Zimmer der Assistenten zu gehen und diese um Rat zu fragen, sondern sie müssen die vorhandenen Kanäle wie Foren oder Email benutzen, die meistens nicht die schnelle Interaktion bieten, wie der Besuch bei den Assistenten.

Obwohl den StudentInnen volle Administrationsrechte auf den Routern gewährt werden und sie diese über eine Shell konfigurieren, wie es real im Labor gemacht wurde, vermissten sie doch den für sie positiven Kontakt zur Hardware, den sie beim traditionellen Labor hatten.

Ein weiterer technischer Punkt ist das Benützen von Java-Applets. Obwohl Javaanwendungen, wie der benutzte SSH-Client, grundsätzlich plattformunabhängig und Browserunabhängig sein sollten, musste festgestellt werden, dass es mit gewissen Netscapeversionen unter Solaris und dem Browser Opera Probleme beim Ausführen des Applets gab.

8.3 Ausblick

Das beschriebene und implementierte Internetportal kann auf mehrere Arten erweitert werden. Einige Ideen seien an dieser Stelle erwähnt.

8.3.1 Auswahl des Router IOS

Erfahrenen StudentInnen könnten verschiedene IOS für die Cisco Router angeboten werden. Um dies zu erreichen, müsste auf dem Portal ein TFTP-Server aufgesetzt werden, der die verschiedenen Versionen des IOS speichert. Das beschriebene Routerresetskript, welches den Router in den Rom-Mon bringt, könnte so angepasst werden, dass es nicht das Konfigurationsregister umstellt, sondern die entsprechenden Befehle an den Router sendet, dass dieser ein gewähltes IOS-Image vom Portal herunterlädt. Nach einem geskripteten Neustart des Routers wäre dieser zur Benutzung mit einem anderen IOS bereit.

8.3.2 Zusätzliche Konfiguration der Linux-Rechner

Im Moment ist die Netzwerkkonfiguration der Linux-Rechner vorbereitet, weil die StudentInnen diese nur für Messzwecke brauchen. Falls Studenten lernen sollten, die Rechner auf der Basis einer frischen Linux-Installation zu konfigurieren, müsste man ihnen vollen Administrationszugang zu den Rechnern geben. Dies könnte jedoch in einem instabilen oder zerstörten System resultieren. Daher müsste eine Möglichkeit gefunden werden, um auch die Linux-Rechner in einen definierten Zustand zu bringen.

Dies könnte erreicht werden, indem man auch die Stromzufuhr der Linux-Rechner via die Relaiskarte steuert und somit diese beliebig ausschalten kann. Das BIOS der Rechner müsste dann so eingestellt werden, dass sie vom eingebauten CD- oder Diskettenlaufwerk starten. Dieses startfähige Medium müsste ein kleines Linux-System enthalten, welches vom Portal via DHCP eine IP Adresse bezieht und mittels TFTP eine vorbereitete Imagedatei herunterlädt. Diese würde auf dem Rechner extrahiert und das darin enthaltene System installiert. Nach einem Neustart wäre der Rechner wieder bereit.

8.3.3 Automatisieren der Kontrolle der Messresultate

Zu diesem Zeitpunkt geben die StudentInnen die Ausgabe der Messwerkzeuge wie Tcpdump und Netpipe den Assistenten zur Kontrolle ab. Interessant wäre ein Mechanismus, der diese Dumps automatisch kontrolliert. Mittels Erweiterung des existierenden Resetskriptes könnten die Konfigurationsdateien der Router auf den TFTP-Server des Internetportals kopiert werden. Diese müssten dann auf dem Portal geparkt werden, um zu kontrollieren, ob alle Konfigurationsschritte vorhanden und in der richtigen Reihenfolge sind.

Leider müsste dieser Parser für jedes andere IOS und für verschiedene Übungen angepasst werden. Um die Ausgabe der Messwerkzeuge von den Linux-Rechnern auf das Portal zu kopieren, könnte man den Befehl „rcp“ benutzen, doch der Parser ist immer noch stark von der Art der Messungen und der benützten Werkzeuge abhängig.

8.3.4 Anpassen des Internetportals für andere Geräte

Das Anbieten von realen Geräten für den Fernzugriff via Internet ist die Hauptaufgabe des Internetportals. Es liegt daher auf der Hand, dass man ausser den beschriebenen Geräten wie Router und Linux-Rechner, auch andere Geräte wie zum Beispiel Steuerungs- oder Messeinrichtungen an das Portal anbinden möchte. Sind diese mit Netzwerk- (Ethernet), seriellen oder parallelen (Druckeranschluss) Schnittstellen ausgestattet, können sie direkt mit dem Portal verbunden werden. Besitzen die anzuschliessenden Geräte jedoch proprietäre Schnittstellen, müsste zuerst eine Möglichkeit gefunden werden, um diese Schnittstellen mit den vorhandenen des Portals zu verbinden.

Ist das Gerät mit dem Portal physikalisch verbunden, muss als erstes ein neuer Benutzer auf dem Portal erstellt werden (siehe auch Kapitel 6.2.3). Kann dieses Gerät mittels einer Linux-Kommandozeilenapplikation wie zum Beispiel Minicom konfiguriert und benutzt werden, muss diese auf dem Portal für den angelegten Benutzer installiert werden. Anschliessend wird dessen Loginskript (siehe auch Kapitel 7.1.1) so angepasst, dass beim Anmelden dieses Benutzers die Applikation gestartet wird. Danach muss im PHP-Code des Webfrontends noch ein neuer Link erstellt werden und das Gerät kann für den Fernzugriff angeboten werden, indem die Geräteapplikation im SSH-Applet betrieben wird.

Schwieriger wird es, wenn ein Gerät oder eine Einrichtung zwar eine passende Schnittstelle zum Portal besitzt, aber keine Applikationen oder Treiber für Linux verfügbar sind. In diesem Fall ist der Aufwand höher, denn es muss (wie bei der Relaiskarte, siehe auch Kapitel 7.2.2.2) Software implementiert werden, mittels der das Portal das Gerät steuern kann. Ist für die Steuerung eines solchen Gerätes, zum Beispiel eines Mikroskops, die Kommandozeilenanbindung ungeeignet und eine grafische Benutzeroberfläche wünschenswert, müsste das SSH-Applet durch ein anderes Applet ersetzt werden. Dieses Applet kann über das Internet mit dem Internetportal verbunden werden und Benutzereingaben an das Portal schicken. Die empfangenen Eingaben und Steuerungsanweisungen werden von der portalseitigen Software ausgewertet und entsprechende Steuerungssignale an das Gerät geschickt, welches im Falle des Mikroskops z.B. den sichtbaren Ausschnitt verschiebt oder den Fokus verändert. Rückmeldungen des Gerätes werden an das Portal geschickt, welches diese auswertet und im Applet des Browsers des Benutzerrechners darstellt.

9 Anhang

9.1 Implementierter Quellcode

9.1.1 Die Relaiskartentreiberbibliothek RelaisLib.pm

```
#####  
#  
# gateway for a remote laboratory  
#  
#  
# file:      RelaisLib.pm  
#  
# purpose: simple perl api to control the serial line 8-port relais card  
#           from conrad. with this api we will powercycle our two cisco  
#           routers to delete the set passwords on our cisco routers  
#  
# depends: on the module Device:SerialPort  
#  
# author:  stefan.zimmerli@iam.unibe.ch  
#  
#  
#####  
  
package RelaisLib;  
  
use strict;  
use Device::SerialPort qw( :PARAM :STAT 0.07);  
  
$RelaisLib::RELAIS_CARD_SERIAL_DEVICE= "/dev/ttyS2";  
$RelaisLib::NOP      = 0;  
$RelaisLib::INIT     = 1;  
$RelaisLib::GET      = 2;  
$RelaisLib::SET      = 3;  
$RelaisLib::ADDR     = 1;  
$RelaisLib::C2600    = 0;  
$RelaisLib::C3600    = 1;  
$RelaisLib::POWER_OFF= 0;  
$RelaisLib::POWER_ON = 1;  
  
# debugging/logging of the library functions to standard output  
# 0: logging, 1: no logging  
# set them here OR in the perl program that uses this library!  
#  
#$RelaisLib::NODEBUG= 0;  
#$RelaisLib::NODEBUG= 1;  
  
#####  
#  
#####  
sub get_relais_status  
{
```

```

# input:  tty got from relaiscard_tty_open
# output: relais status in decimal (0: all relais off, 1: relais 0 on, 2:
relais 1 on, 255: all relais on)

my $mytty = shift;

#print "\n==== INIT CARD ==== \n";

RelaisLib::relaiscard_send_command($mytty,$RelaisLib::ADDR,$RelaisLib::INIT,0);
sleep (1);

#print "\n==== GET PORT STATUS ==== \n";
my @portstatus=
RelaisLib::relaiscard_send_command($mytty,$RelaisLib::ADDR,$RelaisLib::GET,0);
#print "port status=$portstatus[2] \n";

$portstatus[2];

}

#####
#
#####
sub relaiscard_send_command
{
# input:  tty,card_address,command,parameter
# output: decoded answer frame from relaiscard

print "---- sub relaiscard_send_command starts ---- \n" unless
$RelaisLib::NODEBUG;
my ($mytty,$card_address,$command,$parameter)= @_;

my @answer= 0;

my $data= encode_frame($command,$card_address,$parameter);
my $count_out= $mytty->write($data);
warn "write failed \n" unless ($count_out);
warn "write incomplete \n" if ($count_out != length($data));

sleep(1);
my ($count_in,$input)= $mytty->read(4);
if (check_frame($input))
{
@answer= decode_frame($input);
print "frame bytes \tsent\treceived \n" unless $RelaisLib::NODEBUG;
print "command \t$command\t$answer[0] \n" unless $RelaisLib::NODEBUG;
print "cardaddr \t$card_address\t$answer[1] \n" unless $RelaisLib::NODEBUG;
print "data \t\t$parameter\t$answer[2] \n" unless $RelaisLib::NODEBUG;

}

if ($command==$RelaisLib::INIT)
{

my ($count_in_init,$input_init)= $mytty->read(4);
if (check_frame($input_init))
{
my @second_init_frame= decode frame($input_init);

```

```

        #print "second init frame @second_init_frame\n";
    }

}

    print "---- sub relaiscard_send_command ends ----\n" unless
$RelaisLib::NODEBUG;

    @answer;
}

#####
#
#####
sub relaiscard_tty_close
{
    # input:  the port handle got from relaiscard_tty_open
    # output: void

    my $myport= shift(@_);
    $myport->close || die "failed to close $!\n";
    undef $myport;
}

#####
#
#####
sub relaiscard_tty_open
{
    # input:  the serial device where the relais card is attached to
(/dev/ttyS2)
    # output: the port handle

    my $device= shift;
    my $myport= new Device::SerialPort($device) || die "cant open $device
$!\n";
    $myport->user_msg("ON");
    $myport->databits(8);
    $myport->baudrate(19200);
    $myport->parity("none");
    $myport->stopbits(1);
    $myport->write_settings || undef $myport;

    $myport;
}

#####
#
#####
sub encode_frame
{
    # input:  three integers
    # output: four-byte-string with checksum

    print "---- sub encode frame starts ----\n" unless $RelaisLib::NODEBUG;

```

```

my($byte0,$byte1,$byte2)= @_;
my $checksum = $byte0 ^ $byte1 ^ $byte2;

my $frame= pack("CCCC",$byte0,$byte1,$byte2,$checksum);

print "encoded data=($byte0/$byte1/$byte2/$checksum), packed
data=<$frame>\n" unless $RelaisLib::NODEBUG;

print "---- sub encode_frame ends ----\n" unless $RelaisLib::NODEBUG;

$frame;
}

#####
#
#####
sub decode_frame
{
    # input:  four-byte-string with checksum
    # output: array with three integers (command, address, data)

    my($frame)= shift (@_);
    my($frame_length)= length($frame);
    #print "---- sub decode_frame starts ----\n" unless $RelaisLib::NODEBUG;
    #print "frame=$frame, length=$frame_length\n" unless $RelaisLib::NODEBUG;

    my @data= unpack("CCCC",$frame);
    #print "@data\n" unless $RelaisLib::NODEBUG;
    #print "---- sub decode_frame ends ----\n" unless $RelaisLib::NODEBUG;

    @data;
}

#####
#
#####
sub check_frame
{
    # input:  four-byte-string with checksum
    # output: 1 if checksum is ok
    #         0 otherwise
    my($in_frame)= shift (@_);
    my($frame_length)= length($in_frame);

    my @frame= unpack("CCCC",$in_frame);

    if (($frame[0] ^ $frame[1] ^ $frame[2]) == $frame[3])
    {
        #print "frame ok\n" unless $RelaisLib::NODEBUG;
        return 1;
    }
    else
    {
        #print "frame NOT ok\n" unless $RelaisLib::NODEBUG;
        return 0;
    }
}

```

```
}

#####
#
#####
# required that require or use "RelaisLib" works

1;
```

Datei 7: Relaiskartentreiber RelaisLib.pm

9.1.2 Die Routerbibliothek RouterLib.pm

```
#####  
#  
# gateway for a remote laboratory  
#  
#  
# file: RouterLib.pm  
#  
# purpose: provides helper functions to send commands to the routers  
#           over the serial line  
#  
# depends: on the module Device:SerialPort  
#  
# author: stefan.zimmerli@iam.unibe.ch  
#  
#  
#####  
  
package RouterLib;  
  
require RelaisLib;  
use Device::SerialPort qw( :PARAM :STAT 0.07);  
  
$RouterLib::LOCKFILE_PATH= '/tmp/';  
$RouterLib::MAX_TIME= 360;  
$RouterLib::CISCO2600_DEVICE= "/dev/ttyS0";  
$RouterLib::CISCO3600_DEVICE= "/dev/ttyS1";  
$RouterLib::CISCO2600_OFF = 254;  
$RouterLib::CISCO2600_ON  = 1;  
$RouterLib::CISCO3600_OFF = 253;  
$RouterLib::CISCO3600_ON  = 2;  
$RouterLib::CISCO2600     = "cisco2600";  
$RouterLib::CISCO3600     = "cisco3600";  
  
#####  
#  
#####  
sub create_lockfile  
{  
    my $mylockfile= $RouterLib::LOCKFILE_PATH;  
    $mylockfile.= shift;  
    print "create: lockfile=$mylockfile\n" unless $RouterLib::NODEBUG;  
    open (LOCKFILE,">".$mylockfile) or  
        die "can not make lockfile $mylockfile: $!\n";  
    close (LOCKFILE);  
}  
  
#####  
#  
#####  
sub delete_lockfile  
{  
    my $mylockfile= $RouterLib::LOCKFILE_PATH;  
    $mylockfile.= shift;  
    print "delete: lockfile=$mylockfile\n" unless $RouterLib::NODEBUG;  
    unlink($mylockfile) or  
        die "can not unlink lockfile $mylockfile: $!\n";  
}
```

```

}
#####
#
#####
sub exist_lockfile
{
    my $mylockfile= $RouterLib::LOCKFILE_PATH;
    $mylockfile.= shift;
    print "exist: lockfile=$mylockfile\n" unless $RouterLib::NODEBUG;
    return (-e $mylockfile);
}

#####
#
#####
sub is_lockfile_new
{
    my $mylockfile= $RouterLib::LOCKFILE_PATH;
    $mylockfile.= shift;
    print "is new: lockfile=$mylockfile\n" unless $RouterLib::NODEBUG;
    my @mylockfile_stats= stat($mylockfile);
    if ((time-$mylockfile_stats[9]) < $RouterLib::MAX_TIME)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

#####
#
#####
sub send_router_command
{
    # input: port (got from "new
Device::SerialPort($cisco2600_device")),
    #         command string
    #         time to sleep in seconds after sending command
    # output: void

    my $_port= shift;
    my $_command= shift;
    my $_time= shift;
    print "command=$_command\n sending it and wait $_time seconds..."
unless $RouterLib::NODEBUG;
    my $_count_out= $_port->write($_command);
    warn "write failed\n" unless ($_count_out);
    warn "write incomplete\n" if ($_count_out != length($_command));
    sleep($_time);
    print "done\n\n" unless $RouterLib::NODEBUG;
}

#####
#
#####
sub reset_router

```



```

{
    my $myrouter= shift;
    print "reset router $myrouter\n" unless $RouterLib::NODEBUG;

    my $tty=
RelaisLib::relaiscard_tty_open($RelaisLib::RELAIS_CARD_SERIAL_DEVICE);

    my $status= RelaisLib::get_relais_status($tty);
    my $myport= '';

    if ($myrouter eq $RouterLib::CISCO2600) {

        my $new_status= $status & $RouterLib::CISCO2600_OFF;
        RelaisLib::relaiscard_send_command(
            $tty,$RelaisLib::ADDR,$RelaisLib::SET,$new_status);
        sleep(2);

        my $power_on = $new_status | $RouterLib::CISCO2600_ON;

        RelaisLib::relaiscard_send_command(
            $tty,$RelaisLib::ADDR,$RelaisLib::SET,$power_on);

        RelaisLib::relaiscard_tty_close($tty);
        sleep(13);

        $myport= new Device::SerialPort($RouterLib::CISCO2600_DEVICE)
            || die "cant open $RouterLib::CISCO2600_DEVICE $!\n";

        $myport->user_msg("ON");
        $myport->databits(8);
        $myport->baudrate(9600);
        $myport->parity("none");
        $myport->stopbits(1);
        $myport->write_settings || undef $myport;

        $myport->pulse_break_on(1000);

        send_router_command($myport,"confreg 0x2142\n",1);
        send_router_command($myport,"reset\n",106);

        send_router_command($myport,"no\n",2);
        send_router_command($myport,"\r",2);
        send_router_command($myport,"\r",2);

        send_router_command($myport,"enable\n",1);

        send_router_command($myport,"configure\n",1);
        send_router_command($myport,"terminal\n",1);
        send_router_command($myport,"config-register 0x2102\n",1);
        send_router_command($myport,"exit\n",1);
        send_router_command($myport,"copy running-config startup-
config\n",1);
        send_router_command($myport,"startup-config\n",12);
        send_router_command($myport,"reload\n",1);
        send_router_command($myport,"\n",106);
        send_router_command($myport,"\r",2);
    }
    if ($myrouter eq $RouterLib::CISCO3600) {

        my $new_status= $status & $RouterLib::CISCO3600_OFF;

```

```

        RelaisLib::relaiscard_send_command(
            $tty,$RelaisLib::ADDR,$RelaisLib::SET,$new_status);
sleep(2);

my $power_on = $new_status | $RouterLib::CISCO3600_ON;

        RelaisLib::relaiscard_send_command(
            $tty,$RelaisLib::ADDR,$RelaisLib::SET,$power_on);

        RelaisLib::relaiscard_tty_close($tty);

sleep(16);

$myport= new Device::SerialPort($RouterLib::CISCO3600_DEVICE) ||
die "cant open $RouterLib::CISCO3600_DEVICE $!\n";

$myport->user_msg("ON");
$myport->databits(8);
$myport->baudrate(9600);
$myport->parity("none");
$myport->stopbits(1);
$myport->write_settings || undef $myport;

$myport->pulse_break_on(1000);

send_router_command($myport,"confreg 0x2142\n",1);

send_router_command($myport,"reset\n",100);

send_router_command($myport,"no\n",2);
send_router_command($myport,"\r",2);
send_router_command($myport,"\r",2);

send_router_command($myport,"enable\n",1);

send_router_command($myport,"configure\n",1);
send_router_command($myport,"terminal\n",1);
send_router_command($myport,"config-register 0x2102\n",1);
send_router_command($myport,"exit\n",1);
send_router_command($myport,"copy running-config startup-
config\n",1);
send_router_command($myport,"startup-config\n",1);

send_router_command($myport,"reload\n",1);
send_router_command($myport,"\n",120);
send_router_command($myport,"\r");
}

$myport->close || die "failed to close $device";
undef $myport;

}
#####
#
#####
1;

```

Datei 8: Routerbibliothek RouterLib.pm

9.1.3 Das Routerresetskript pw_reset.pl

```
#!/usr/bin/perl
#####
#
# gateway for a remote laboratory
#
#
# file:    pw_reset.pl
#
# purpose: resets the cisco routers and logs the action in the syslog
#
# depends: on the module Device:SerialPort, RouterLib.pm, RelaisLib.pm
#
# author:  stefan.zimmerli@iam.unibe.ch
#
#
#####
use Sys::Syslog;
require RelaisLib;
require RouterLib;

$RelaisLib::NODEBUG= 1;
$RouterLib::NODEBUG= 1;

openlog('pw_reset ', 'pid', 'syslog');

my $router= shift;
my $RouterResetCmd= '';
my $lockfile= '';

syslog('debug', "pw_reset.pl start");
syslog('debug', "input=$router\n");

if (($router eq $RouterLib::CISCO2600) || ($router eq
$RouterLib::CISCO3600)) {

    syslog('debug', "router=$router\n");

    $lockfile= $router;

    if (RouterLib::exist_lockfile($lockfile))
    {
        syslog('debug', "lockfile $lockfile exists\n");

        if (RouterLib::is_lockfile_new($lockfile))
        {
            syslog('debug', "lockfile $lockfile is new. reset in
progress!\n");
        }
        else
        {
            syslog('debug', "2 lockfile $lockfile old deleting it\n");
            RouterLib::delete_lockfile($lockfile);

            syslog('debug', "2 creating lockfile $lockfile\n");
            RouterLib::create_lockfile($lockfile);

            syslog('debug', "2 performing reset\n");
        }
    }
}
```

```

RouterLib::reset_router($router);
syslog('debug',"2 reset done\n");

syslog('debug',"2 deleting lockfile $lockfile\n");
RouterLib::delete_lockfile($lockfile);
}

}
else
{
syslog('debug',"1 creating lockfile $lockfile\n");
RouterLib::create_lockfile($lockfile);

syslog('debug',"1 performing reset\n");
RouterLib::reset_router($router);
syslog('debug',"1 reset done\n");

syslog('debug',"1 deleting lockfile $lockfile\n");
RouterLib::delete_lockfile($lockfile);
}
}

syslog('debug',"pw_reset done");
closelog;

```

Datei 9: Routerresetskript pw_reset.pl

9.2 Referenzen

- [1] Schweizerisches Bundesamt für Bildung und Wissenschaft, <http://www.bbw.admin.ch/>
- [2] Swiss Virtual Campus (SVC), <http://www.virtualcampus.ch/>
- [3] Virtual Internet and Telecommunications Laboratory of Switzerland (VITELS), <http://www.vitels.ch/>
- [4] WebCT, <http://www.webct.com>
- [5] Jampen: Authentication, Authorization and Resource Reservation for Distributed Laboratories, Diplomarbeit, 2002
- [6] Institut für Informatik und angewandte Mathematik (IAM), <http://www.iam.unibe.ch>
- [7] Universität Bern, <http://www.unibe.ch>
- [8] Cisco Router, <http://www.cisco.com>
- [9] Steinemann, Jampen, Zimmerli, Braun: Didactical Issues of a Remote Network Laboratory, 4th International Conference on New Educational Environments (ICNEE 02), Lugano, May 8-11, 2002, ISBN 3-0345-0031-9, pp. 1.2/39-41
- [10] Steinemann, Braun: Remote versus Traditional Learning in a Computer Networks Laboratory, Communications and Computer Networks (CCN 2002), Cambridge, USA, November 4-6, 2002, ISBN 0-88986-329-6, pp. 503-507
- [11] SVC Projekte, <http://www.virtualcampus.ch/display.php?lang=1&zid=65>
- [12] Distance learning, E-Learning <http://www.brandonhall.com/public/glossary/glossary.html#Distance Learning>
- [13] W3 Schools, <http://www.w3schools.com/default.asp>
- [14] Mentor Technologies, <http://www.mentortech.com.sg/>
- [15] Emulab, <http://www.emulab.net>
- [16] Netzwerksimulator ns-2, <http://www.isi.edu/nsnam/ns/>
- [17] Nanoworld, <http://www.nanoworld.unibas.ch/nano>
- [18] Ariadne, <http://ariadne.unil.ch/>
- [19] Top Class E-Learning Suite, <http://www.wbtsystems.com/>
- [20] Lotus Learning Space, <http://www.lotus.com/products/learnspace.nsf/wdocs/homepage>
- [21] Linux, <http://www.linux.org/>
- [22] Open-Source, <http://www.opensource.org/>
- [23] UNIX, <http://www.opengroup.org/>
- [24] Apache, <http://www.apache.org/>
- [25] NCSA httpd, <http://hohoo.ncsa.uiuc.edu/>
- [26] PHP Hypertext Preprocessor, <http://www.php.net/>
- [27] MySQL, <http://www.mysql.com/>
- [28] PostgreSQL, <http://www.postgresql.org/>
- [29] Oracle, <http://www.oracle.com/>
- [30] PERL, <http://www.perl.com>
- [31] Common Gateway Interface (CGI), <http://hohoo.ncsa.uiuc.edu/cgi/overview.html>

- [32] LDAP, <http://www.ietf.org/rfc/rfc1777.txt>
- [33] X.509 Protokoll, <http://www.ietf.org/rfc/rfc1798.txt>
- [34] OpenLDAP, <http://www.openldap.org/>
- [35] Secure Sockets Layer Protokoll (SSL),
<http://wp.netscape.com/eng/ssl3/>
- [36] Transport Control Protocol (TCP), <http://www.rfc-editor.org/rfc/rfc793.txt>
- [37] Hypertext Transport Protocol (HTTP),
<http://www.w3.org/Protocols/>
- [38] File Transfer Protocol (FTP), <http://www.w3.org/Protocols/rfc959/>
- [39] Telnet, <http://www.faqs.org/rfcs/rfc854.html>
- [40] Transport Layer Security (TLS), <http://www.ietf.org/rfc/rfc2246.txt>
- [41] OpenSSL, <http://www.openssl.org/>
- [42] Hyper Text Transport Protocol over Secure Sockets Layer (HTTPS),
<http://developer.netscape.com/docs/manuals/security/sslin/contents.htm>
- [43] Microsoft Internet Information Server (IIS),
<http://www.microsoft.com/windows2000/technologies/web/default.asp>
- [44] mod_ssl, <http://www.modssl.org/>
- [45] Secure Shell (SSH), <http://www.ssh.com/>
- [46] OpenSSH, <http://www.openssh.com/>
- [47] Mindterm, <http://www.appgate.com/mindterm/>
- [48] Java, <http://java.sun.com/>
- [49] Applet, <http://java.sun.com/applets/>
- [50] Sandbox, <http://www.securingsjava.com/chapter-two/>
- [51] Java Virtual Machine, <http://java.sun.com/docs/books/vmspec/>
- [52] Internet Engineering Task Force (IETF), <http://www.ietf.org>
- [53] Ipv6, <http://www.ipv6.org/>
- [54] FreeS/wan, <http://www.freeswan.org/>
- [55] VPN, <http://www.vpnc.org/>
- [56] Steinemann, Jampen, Zimmerli, Braun: Architectural Issues of a Remote Network Laboratory, Networked Learning 2002 (NL 2002), Berlin, May 1-4, 2002, ISBN 3-906454-31-2, CD-ROM
- [57] Steinemann, Zimmerli, Jampen, Braun: Global Architecture and Partial Prototype Implementation for Enhanced Remote Courses, Computers and Advanced Technology in Education (CATE 2002), Cancun, Mexico, May 20-22, 2002
- [58] Informatikdienste der Universität Bern, <http://www.id.unibe.ch/>
- [59] Ping Command Reference.
<http://www.cisco.com/univercd/cc/td/doc/product/cddi/c1100/1100dp/58808.htm#xtocid2017134>
- [60] Traceroute Command,
http://www.cisco.com/univercd/cc/td/doc/product/atm/ls1010s/wa4/11_3/trb_gd/tools.htm#xtocid5
- [61] INTERNET CONTROL MESSAGE PROTOCOL (ICMP),
<http://www.ietf.org/rfc/rfc0792.txt>
- [62] Netpipe, A Network Protocol Independent Performance Evaluator,
<http://www.scl.ameslab.gov/netpipe/>
- [63] Tcpcat, <http://www.tcpcat.org/>

- [64] Network Information System (NIS), <http://www.sun.com/>
- [65] Remote Procedure Calls (RPC),
<http://www.cs.cf.ac.uk/Dave/C/node33.html>
- [66] pam_ldap, http://www.padl.com/OSS/pam_ldap.html
- [67] Pluggable Authentication Module (PAM),
<http://www.kernel.org/pub/linux/libs/pam/>
- [68] BSD Rlogin, <http://www.faqs.org/rfcs/rfc1258.html>
- [69] Chroot Login Howto, <http://tjw.org/chroot-login-HOWTO/>
- [70] SLSNIF, <http://www.azstarnet.com/~ymg/software.html>
- [71] Cisco password recovery solution,
<http://www.cisco.com/warp/public/474/>
- [72] Trivial File Transfer Protocol (TFTP),
<http://www.ietf.org/rfc/rfc1350.txt>
- [73] 8-Fach Relaiskarte, Conrad Elektronik, Solothurn, Bestell-Nr. 96 77
20-22
- [74] Steuerungsprotokoll des Relaiskarten-Mikrokontrollers, Conrad 8-
Fach Relaiskartenhandbuch, Seiten 5-8
- [75] Device::SerialPort, Perl-Modul,
<http://www.perldoc.com/cpan/Device/SerialPort.html>

9.3 Abbildungsverzeichnis

Abbildung 1: Virtuelles Privates Netzwerk	15
Abbildung 2: Aufbau des traditionellen IPSec Moduls	17
Abbildung 3: IP-Adress-Konfiguration des Routers 2600.....	20
Abbildung 4: RIP-Konfiguration des Routers 2600	21
Abbildung 5: IP-Adress-Konfiguration des Routers 3600.....	21
Abbildung 6: RIP-Konfiguration des Routers 3600	21
Abbildung 7: Pingen des host3 vom host1 aus	22
Abbildung 8: Traceroute vom host1 zum host3.....	22
Abbildung 9: Pingen des host1 vom host3 aus	22
Abbildung 10: Traceroute vom host3 zum host1	23
Abbildung 11: Verbindung zweier Netze mittels VPN-Tunnel.....	24
Abbildung 12: Generieren der Schlüssel	25
Abbildung 13: Austauschen der Schlüssel.....	26
Abbildung 14: Konfigurieren der Router.....	26
Abbildung 15: : Bandbreitenmessung ohne VPN-Tunnel (Empfänger).....	27
Abbildung 16: Bandbreitenmessung ohne VPN-Tunnel (Sender).....	27
Abbildung 17: Bandbreitenmessung mit VPN-Tunnel (Empfänger)	28
Abbildung 18: Bandbreitenmessung mit VPN-Tunnel (Sender).....	28
Abbildung 19: Tcpdump: Mitlesen des Verkehrs vom Rechner host1 zum Rechner host3.....	29
Abbildung 20: Telnet vom host1 zum host3.....	29
Abbildung 21: Tcpdump-Ausgabe ohne VPN-Tunnel	30
Abbildung 22: Tcpdump-Ausgabe mit VPN-Tunnel.....	31
Abbildung 23: Die VITELS-Architektur	33
Abbildung 24: VITELS-Kursmodule und Modulreservierung im Kurssystem WebCT	34
Abbildung 25: Administrationsansicht der Studentendaten im Kurssystem WebCT..	35
Abbildung 26: Verzeichnisstruktur für Universitäten.....	36
Abbildung 27: Verzeichnisstruktur für VITELS-spezifische Einträge.....	37
Abbildung 28: Das Reservationssystem	38
Abbildung 29: Das Portal als Hub	43
Abbildung 30: Das Portal als Firewall.....	44
Abbildung 31: Anbindung des Internetportals an die VITELS-Architektur	46
Abbildung 32: Anmeldemaske des IPSec-Moduls	49
Abbildung 33: Fehlerseite des IPSec-Moduls.....	50
Abbildung 34: Hauptseite (Labor) des IPSec-Moduls.....	51
Abbildung 35: Abmeldeseite des IPSec-Moduls	52
Abbildung 36: Zugriff auf den Router cisco2600 mittels SSH-Applet	53
Abbildung 37: Zugriff auf den Rechner host1 mittels SSH-Applet	54
Abbildung 38: Internetportal und Laborgeräte des VITELS-Fernkursmoduls IPSec ..	55
Abbildung 39: Anmeldeweiterleitung an Linux-Rechner und Cisco Router.....	56
Abbildung 40: Dateisystemstruktur der Change-Root-Umgebung.....	57
Abbildung 41: Anmeldungsablauf des Chroot-Root-Benutzers host2	58
Abbildung 42: Routermeldung nach Einschalten	62
Abbildung 43: Routermeldung nach Break-Sequenz	62
Abbildung 44: Umstellen des Konfigurationsregisters.....	62

Abbildung 45: Routermeldung nach Neustart	63
Abbildung 46: Zurückstellen des Konfigurationsregisters	63
Abbildung 47: Routermeldung nach zweitem Neustart.....	64
Abbildung 48: Ablauf des Skriptes pw_reset.pl	67

9.4 Tabellenverzeichnis

Tabelle 1: Kursmodule des VITELS-Basiskurses	2
Tabelle 2: Zuweisung der IP-Adressen.....	18
Tabelle 3: Routingtabelle der Linux-Rechner.....	19
Tabelle 4: Attribute der Timetable-Einträge.....	39
Tabelle 5: Attribute der Timeslot-Einträge.....	39
Tabelle 6: Attribute der Module-Einträge.....	39
Tabelle 7: Kodierung der Befehls- und Antwortrahmen	64

9.5 Verzeichnis der Konfigurations- und der Quellcodedateien

Datei 1: Loginskript des Portalusers host1	56
Datei 2: rhost Konfiguration des Users host1 auf dem Linux-Rechner host1	56
Datei 3: Loginskript des Portalusers cisco2600	57
Datei 4: /etc/passwd-Einträge der Portalbenutzer host1, host2 und host3	58
Datei 5: Shellskript /bin/chroot-shell des Benutzers host2	58
Datei 6: sudo-Konfigurationsdatei /etc/sudoers	59
Datei 7: Relaiskartentreiber RelaisLib.pm	75
Datei 8: Routerbibliothek RouterLib.pm	79
Datei 9: Routerresetskript pw_reset.pl	81