INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Mandantenfähige Campus-WLAN-Authentifizierung mit RADIUS

Adrian Klein

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Mandantenfähige Campus-WLAN-Authentifizierung mit RADIUS

Adrian Klein

Aufgabensteller:	Priv. Doz. Dr. Wolfgang Hommel
Betreuer:	Jochen Gebert Stefan Metzger Helmut Tröbs
Abgabetermin:	25. April 2014

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

München, den 25. April 2014

.....
(Unterschrift des Kandidaten)

Abstract

In dieser Arbeit wird das Problem der unzureichend gesicherten Zugänge zum Münchner Wissenschaftsnetz aufgegriffen. Diese Zugänge werden häufig von Instituten der Münchner Hochschulen eingerichtet. Zu Beginn wird daher ein kurzer Überblick über die notwendigen Grundlagen von WLAN-Sicherheit und der vorhandenen Infrastruktur gegeben. Im Bezug auf diese Grundlagen und die potentiellen Einsatzzwecke wird eine methodische Anforderungsanalyse erstellt, um festzustellen, welche Eigenschaften eine Alternative aufweisen muss. Auf Basis dieser Analyse wird ein konkreter Entwurf eines besser gesicherten Systems ausgearbeitet. Dieser Systementwurf wird abschließend implementiert.

Inhaltsverzeichnis

1 Einleitung

Die Relevanz von kabellosen lokalen Netzwerken *(wireless local area networks / WLANs)* hat seit ihrer Entwicklung immer weiter zugenommen. Damit einhergehend ist auch das Bedürfnis nach Datenschutz gewachsen, was gerade in den letzten Monaten deutlich zu beobachten war. Dies soll hier zuerst verdeutlicht werden, danach soll aber auch dargestellt werden, welche Probleme dabei auftreten und was mit der Arbeit erzielt werden soll.

1.1 Motivation

Mobil arbeiten zu können ist durch die Verbreitung von Notebooks, Tablets und Smartphones immer wichtiger geworden. In vielen Berufen wird inzwischen erwartet, dass der Arbeitnehmer auch in der Freizeit erreichbar ist [Umf12]. Dies wird davon unterstützt, dass rund 32 Millionen Menschen in Deutschland einen mobilen Rechner in ihrem Haushalt zur Verfügung haben[Ins13]. Zeitgleich ist die Anzahl der Smartphone-Nutzer im Juni 2013 auf rund 36 Millionen gestiegen, beinahe 10 Millionen mehr als noch ein Jahr zuvor[Com13].

Viele dieser Nutzer haben eine Internetflatrate, um mobil arbeiten oder mit Bekannten in Kontakt bleiben zu können. Diese Flatrates sind jedoch vor allem in Ballungsräumen oft der Anzahl an Nutzern nicht gewachsen, was sich in einem Einbruch der Übertragungsraten bemerkbar macht. Außerdem ist das Volumen bis zu einer Geschwindigkeitsdrosselung oft eher gering, sodass nur eine begrenzte Zeit lang sinnvoll gearbeitet werden kann. Diese Problempunkte hat beispielsweise auch die Stadt München erkannt, die bereits an vier großen, öffentlichen Plätzen[1] kostenloses und offenes WLAN ohne Volumenbeschränkung zur Verfügung stellt, welches auch höhere Übertragungsraten erlaubt. [Por13]

Die öffentlichen Bereiche der Ludwig-Maximilians-Universität München *(LMU)*, der Technischen Universität München *(TUM)* so wie anderer Münchner Hochschulen werden größtenteils vom Leibniz-Rechenzentrum *(LRZ)* über die zwei Service Set Identifier (SSIDs) *'lrz'* und *'eduroam'* mit WLAN versorgt, welches den Studenten und Mitarbeitern der Hochschulen zur Verfügung steht[LR13f]. Die nicht-öffentlichen Bereiche werden aber meist nur auf Anfrage versorgt, da dem LRZ die Kapazitäten und Mittel fehlen, selbstständig alles abzudecken[LR13f]. Die Institute können zwar auch eigene Lösungen bereitstellen, diese haben dann allerdings einige Regeln zu befolgen, um ein konsistentes Sicherheitsmodell zu gewährleisten[LR14b].

1.2 Problemstellung

Die Enthüllungen der Überwachung durch die amerikanische National Security Agency *(NSA)* haben eine breitere Diskussion über Verschlüsselung angefacht, die beispielsweise schon viele

[1] Marienplatz, Odeonsplatz, Sendlinger Tor, Stachus

E-Mail-Anbieter dazu veranlasst hat, ihr Angebot nur noch verschlüsselt anzubieten (vgl. [Goo14], [GMX]).

Die in Kapitel 1.1 erwähnten, von der Stadt München bereitgestellten, WLAN-Zugangspunkte werden jedoch nicht verschlüsselt, sondern komplett offen betrieben [Por13]. Dies hat zur Folge, dass leichter Pakete mitgeschnitten und somit beispielsweise Passwörter ausgelesen werden können.
Damit WLAN-Netzwerke als sicher gelten, müssen sie so konfiguriert sein, dass sie den Sicherheitsstandard Wi-Fi Protected Access II *(WPA2)* erfüllen, welcher zwei Betriebsmodi hat. Im privaten Modus gibt es eine Passphrase *(Pre-shared Key / PSK)*, die für alle Clients gleich ist und diesen zum Einwählen mitgeteilt werden muss. Der andere Modus ist für Firmen gedacht und wird daher als Enterprise-Modus bezeichnet. Dieser verwendet Benutzer-Passwort-Kombinationen zur Authentisierung, benötigt jedoch einen RADIUS-Server *(Remote Authentication Dial In User Service)*, der die Anmeldeversuche überprüft.

Bei PSK wird ein einheitliches Passwort für alle Nutzer bereitgestellt, sodass neue Mitarbeiter und Studenten nicht explizit zugelassen werden müssen, um sich mit dem WLAN-Netzwerk zu verbinden. Der Nachteil an dieser Methode ist jedoch, dass beim Ausscheiden eines Mitarbeiters oder Studenten der zentrale Schlüssel geändert werden sollte, um weiterhin nur autorisierten Personen den Zugriff zu ermöglichen. Dies hätte wiederum zur Folge, dass allen Benutzern der neue Schlüssel mitgeteilt werden muss.
Besser ist in diesem Umfeld also die Authentisierung über den Standard 802.1X, welcher den Enterprise-Modus definiert, in Kombination mit einem RADIUS-Server. Das Einrichten und der Betrieb eines eigenen RADIUS-Servers ist für viele Institute jedoch zu komplex.
Viele greifen deshalb auf PSK zurück, was aber nicht den Sicherheitsvorstellungen des LRZ für das Münchner Wissenschaftsnetz *MWN* entspricht[LR14b]. Da das gesamte MWN im nicht-öffentlichen Raum jedoch primär dezentral verwaltet ist, ist die Überprüfung der Sicherheitsrichtlinien nur unter großem Aufwand möglich, da jedes Institut einzeln geprüft werden müsste.

1.3 Ziel der Arbeit

Bei 802.1X bekommt jeder Nutzer eine eigene Kennung, die zwar explizit freigeschaltet werden muss, welche dafür aber einfach einzeln wieder entfernt werden kann. Dies bietet theoretisch auch die Möglichkeit, den Nutzer abhängig von seiner Kennung in verschiedene VLANs zuzulassen.

Im Rahmen der Arbeit soll deshalb ein mandantenfähiges Konzept entwickelt werden, wie die aktuelle Situation, also dass Institute einen Pre-Shared Key verwenden, zu verbessern ist. Mandantenfähig bedeutet hier, dass jeder Kunde in seiner eigenen Datenumgebung arbeitet, nur die eigenen Benutzer verwalten kann und nur diese sich an den entsprechenden Stellen anmelden können. Dies umfasst eine einsatzfähige Installation eines RADIUS-Servers, welcher an die Anforderungen des Münchner Wissenschaftsnetz angepasst wurde. Diese soll so aufbereitet werden, dass sie leicht vervielfältigt werden kann, was z.B. über ein VM-Template oder eine Anleitung geschehen könnte. Am Ende soll auch eine Anleitung entstehen, wie der RADIUS-Server zu installieren und zu konfigurieren ist und welche Schritte vom LRZ und welche von Institutsmitarbeitern notwendig sind. Außerdem soll eine Anleitung erstellt werden, wie die Mitarbeiter der Institute ihre Nutzer verwalten können.

Dies alles soll dazu dienen, dass LRZ und Institute sich leichter auf eine Sicherheitsbasis einigen können und diese leichter realisiert werden kann. Außerdem wird so ermöglicht, dass Gäste an einem Institut schneller und einfacher einen sicheren Zugriff auf lokale Ressourcen oder auch auf das Internet bekommen können, da das Institut selbst die Nutzer hinzufügen kann.

1.4 Struktur der Arbeit

Begonnen wird in Kapitel 2.1 mit der Funktionsweise von WPA2 und dessen gängigster Form der Authentisierung, dem Pre-Shared Key. Das Kapitel 2.2 stellt den Standard 802.1X genauer dar, welcher die Grundlage für den Enterprise-Modus von WPA2 darstellt. Dabei werden in 2.2.1 zuerst zentrale Begriffe für 802.1X eingeführt, bevor in 2.2.2 die Erweiterung durch das EAP-Protokoll näher beleuchtet wird. In 2.2.3 wird dann die Funktionsweise eines RADIUS-Servers und der Kommunikation zwischen diesem und dem WLAN-Router erklärt.

In Kapitel 3 geht es allgemein um die Anforderungen, die ein RADIUS-Server im Hochschulumfeld zu erfüllen hat. Dabei wird zuerst ein Blick auf die aktuelle Infrastruktur des LRZ geworfen und diese danach in 3.2 mit Hilfe weiterer Use-cases auf zu erfüllende Kriterien hin untersucht.

Kapitel 4 wird die genaue Ausarbeitung eines Entwurfs umfassen, angefangen mit der Vorstellung von verschiedenen RADIUS-Servern in 4.2 und gefolgt von der Auswahl eines passenden Betriebssystems für den RADIUS-Server in Kapitel 4.3. Kapitel 4.4 behandelt die Möglichkeiten, wie der RADIUS-Server die Anfragen zugesendet bekommen kann. Weiterhin wird in Kapitel 4.5 entschieden, wie die Benutzerverwaltung geschehen soll. Hierzu werden die Möglichkeiten einer Klartext-Datei, eines LDAP-fähigen Verzeichnisdienstes und einer SQL-Datenbank betrachtet. Zum Schluss des Kapitels wird entschieden, ob dies alles als Template für eine Virtuelle Maschine ausgeliefert wird oder nur eine Anleitung zum Selbst-Einrichten erstellt wird.

In Kapitel 5 wird die praktische Ausführung des vorher erarbeiteten Konzepts dargestellt, beginnend mit einer Beschreibung der gegebenen Testumgebung. In 5.2 wird dann der Vorgang der Installation und Konfiguration der RADIUS-Software beschrieben. Daraufhin werden die verschiedenen durchgeführten Tests und deren Ergebnisse dargestellt. Zum Schluss werden die Anleitungen für die Konfiguration des RADIUS-Proxys, der Installation und Konfiguration des RADIUS-Servers und der Benutzerverwaltung aufgeführt.

2 Technische Hintergründe

Das *Institute of Electrical and Electronics Engineers (IEEE)* hat mit den 802er-Standards eine Sammlung zusammengestellt, die vor allem für lokale Netzwerke *(Local area networks / LANs)* gedacht ist. Für die vorliegende Arbeit sind vor allem zwei dieser Standards wichtig:

- der 802.11i, welcher die Verschlüsselung von WLAN-Netzwerken und die Authentisierung mittels Pre-Shared Keys betrifft.
- der 802.1X, welcher die Port-basierte Authentisierung beschreibt.

2.1 Der Standard 802.11i

Der Standard 802.11i ist eine Erweiterung zum originalen 802.11. Er war zuerst die Basis für WPA und bildet nun in der Fassung von 2004 die Grundlage für WPA2. Dieses erfüllt den Standard jedoch nicht komplett, da Funktionen wie z.B. *Fast Roaming* nicht übernommen wurden[Ele14].

2.1.1 Funktionsweise der Authentisierung bei WPA2 mit PSK

Bei einem Pre-Shared Key sind der Client und der Access-Point *(AP)* in Kenntnis des selben Schlüssels, der out-of-band ausgetauscht werden muss.
Der Ablauf, welcher in Bild 2.1 gezeigt wird, ist wie folgt:

1. Wenn sich die beiden Geräte gefunden haben, sendet der Client eine *Probe Request*-Nachricht an den Access-Point, um Informationen über diesen zu bekommen[Cis12].
2. Der AP liefert die Daten wie beispielsweise unterstützte Übertragungsraten in einer *Probe Response*-Nachricht[Cis12].
3. Nun gibt es zwei Vorgehensweisen, die abhängig davon sind, ob *simultaneous authentication of equals (SAE)* zum Einsatz kommen soll oder nicht.

   SAE basiert auf dem Diffie-Hellman-Verfahren [Wik14b] und wird unter anderem mit dem PSK initialisiert, wodurch eine Authentisierung ohne Versenden des Schlüssels erreicht werden kann [IEE12]. Wenn SAE erfolgreich abgeschlossen wird, dann entsteht hierbei ein neuer Schlüssel, der für die Session eindeutig ist und als *pairwise master key (PMK)* verwendet wird. Ohne SAE wird der Pre-Shared Key direkt als PMK verwendet und es kommt zu einem *Open System Authentication Request* vom Clienten an den AP und einer *Open System Authentication Response* in die andere Richtung. Es wird jedoch davon abgeraten den PSK als PMK zu verwenden, da dies prinzipiell Angriffe zum Knacken der Verschlüsselung erleichtert.

   Der *pairwise master key* wird benötigt, um daraus den *pairwise transient key (PTK)* und den *group transient key (GTK)* abzuleiten. Diese Sichern die Übertragung des

zusätzlich erstellten Temporal Keys, welcher für die Ver- und Entschlüsselung von Nachrichten benötigt wird [HR13a].

4. Daraufhin bittet der Client mit einem *Association Request*, dass der AP Ressourcen für ihn reserviert [Cis12].

5. Der AP teilt dem Client mit einer *Association Response* mit, ob die Anfrage angenommen oder abgelehnt wurde [Cis12].

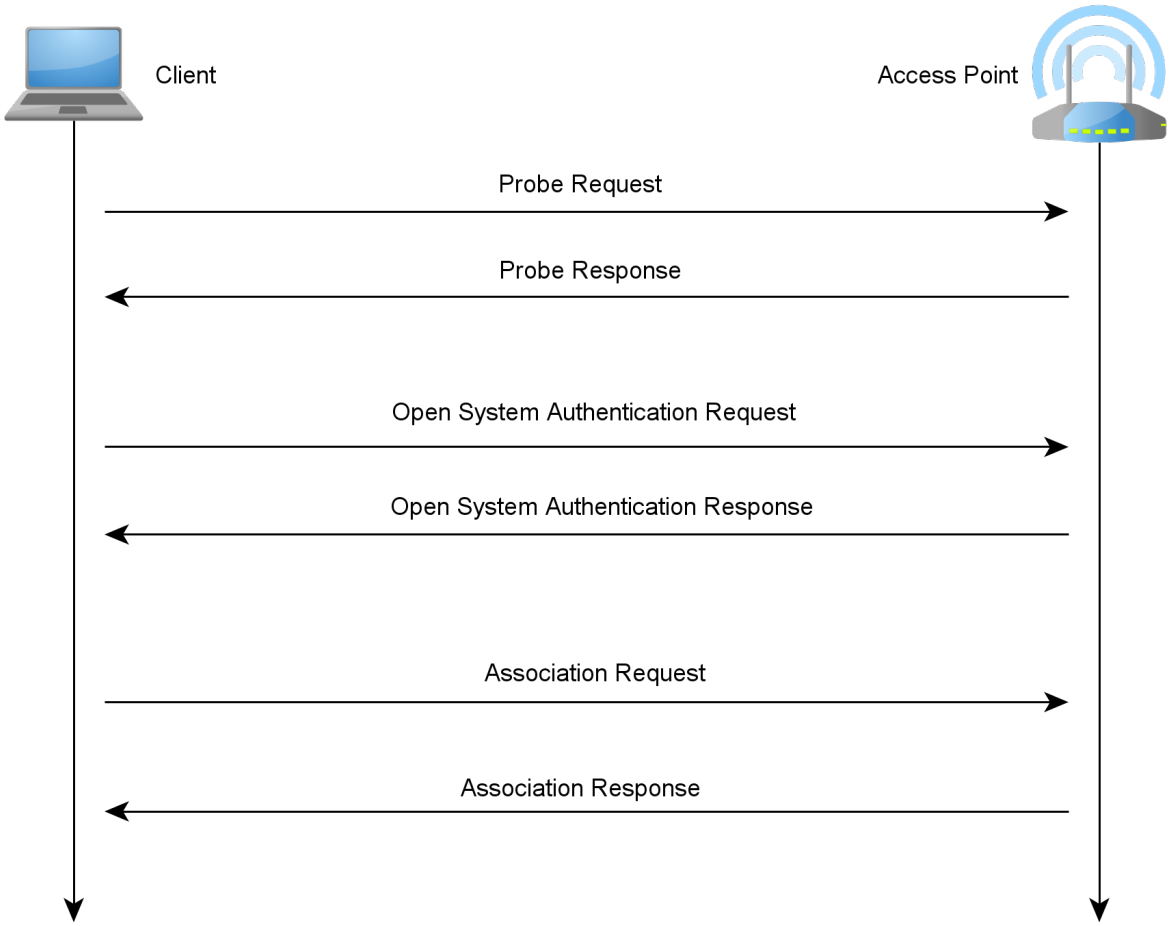

Abbildung 2.1: WPA2 Authentisierung mit einem Pre-Shared Key

2.2 Der Standard 802.1X

Der 802.1X funktioniert seit der Revision von 2004 unabhängig von dem eingesetzten Medium und ist für die gesamte Gruppe der 802er Standards verwendbar, nachdem er ursprünglich nur für Ethernet konzipiert wurde[IEE10]. Er beschreibt, wie die Verwendung des Extensible Authentication Protocols *(EAP)* in einem LAN funktioniert und definiert so ein Protokoll für *EAP over LAN (EAPOL)* [IEE10].

2.2.1 Grundlagen

Abbildung 2.2: Infrastruktur bei 802.1X mit einem Supplicant

Um die Funktionsweise von 802.1X zu erklären werden drei Begriffe eingeführt, welche die Infrastruktur beschreiben:

Als *Supplicant* werden alle Client-Geräte bezeichnet, die 802.1X theoretisch unterstützen, also beispielsweise Laptops und Handys [HR13b]. Diese Funktionalität wird im Regelfall durch entsprechende Software erreicht und hat keine besonderen Hardware-Anforderungen. Supplicants stellen eine Verbindung mit einem sogenannten *Authenticator* her, wenn sie auf Netzwerkressourcen zugreifen wollen. [HR13b] Bei diesem handelt es sich bei WLAN im Regelfall um den Access Point, bei LAN z.B. um einen Switch [HR13b]. Der Authenticator ist die Entität, die eine Authentisierung anfordern kann. Vor der Authentisierung ordnet der Authenticator den Supplicant einem *uncontrolled port* zu, also einer Verbindungsschnittstelle, die nur die Authentisierung erlaubt [HR13b]. Wenn die Authentisierung erfolgreich abgeschlossen wird, dann wird der Supplicant an einen *controlled port* geschaltet, welcher die reguläre Kommunikation zum LAN zulässt [HR13b].

Wenn der Supplicant eine Verbindung mit dem Authenticator aufbaut, kann entweder der Authenticator direkt eine Authentifizierung verlangen, oder der Supplicant fordert zuerst einen controlled port an, woraufhin der Authenticator die Authentifizierung verlangt [HR13b]. Diese Kommunikation zwischen Supplicant und Authenticator läuft über EAPOL ab [IEE10]. Der Aufbau hiervon ist in Abbildung 2.2 dargestellt.

Die Authentisierung selbst wird nicht vom Authenticator übernommen, sondern einem *Authentication Server (AS)* überlassen [IEE10]. Dazu leitet der Authenticator die Daten des Supplicants an den Authentication Server weiter, indem er die über EAPOL gelieferten EAP-Daten vom Supplicant in ein RADIUS-Paket einbettet.

2.2.2 EAP

EAP bietet einen Rahmen zur Authentisierung, der mit verschiedenen Authentisierungsmechanismen funktioniert, wobei die konkrete Wahl erst bei der Kommunikation mit dem Authentication Server getroffen wird. Hierbei dient der Authenticator nur dazu, die Nachrichten zwischen Supplicant und Authentication Server weiterzuleiten [HR13b]. Dies hat den Vorteil, dass der Authenticator nicht ersetzt werden muss, damit neue Authentisierungsmechanismen eingesetzt werden können, da dieser weiterhin nur EAP unterstützen muss. Der Authentication Server dagegen kann leichter aktualisiert werden um diese neuen Mechanismen zu unterstützen. [rfc04] EAP wurde so konzipiert, dass es ohne die Funktionalität des Internet Protokolls auskommt, die Supplicants müssen also nicht über eine IP-Adresse verfügen, um eine Authentisierung durchführen zu können. [rfc04] Diese Notwendigkeit ist

begründet darin, dass die IP-Adresse im Regelfall erst nach der erfolgreichen Authentisierung zugewiesen wird.

2.2.3 RADIUS

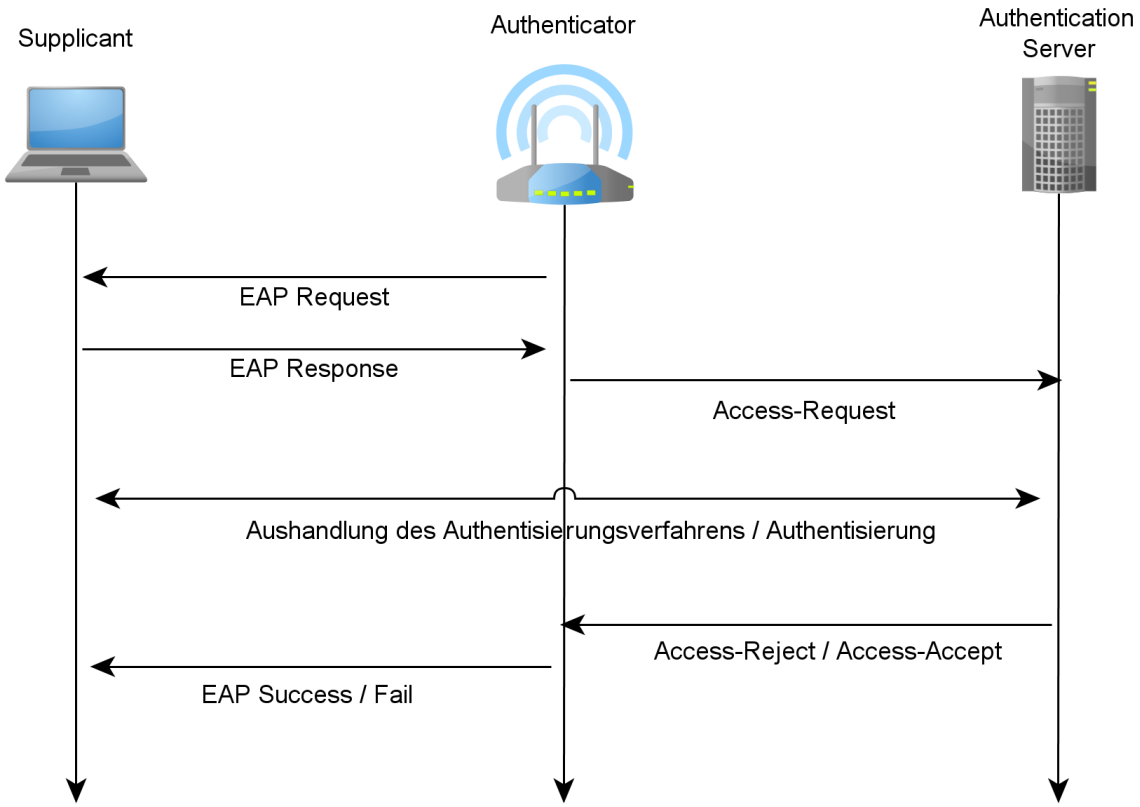

Abbildung 2.3: Zeitleiste einer Anfrage via 802.1X

RADIUS definiert ein Protokoll, welches neben der Authentisierung auch die Autorisierung und Accounting umfasst (*AAA-Protokoll*) [HR13b]. Für die Umsetzung wird das im Kapitel 2.2.2 erläuterte EAP benutzt. Wenn RADIUS im 802.1X-Umfeld verwendet wird, so sendet ein Authenticator eine *Access-Request*-Nachricht an den Authentication Server [Cis06]. Diese Anfrage enthält je nach Verwendung eine Kombination aus Benutzername und Passwort, ein Zertifikat und optional noch weitere Informationen des Supplicants [Cis06]. Der Authentication Server sucht dann in einer lokalen oder ausgelagerten Datenbank nach dem Benutzernamen und überprüft die mitgelieferten Identifikatoren. Wenn der Benutzername nicht gefunden wird oder aber z.B. das Passwort falsch ist, so sendet er eine *Access-Reject*-Nachricht zurück an den Authenticator, optional auch mit einer Begründung für das Ablehnen. [Cis06] Wenn die Daten korrekt verifiziert werden können, dann sendet er stattdessen eine *Access-Accept*-Nachricht, an die Parameter wie beispielsweise Zugriffsrechte, ein bestimmter IP-Adressbereich oder maximale Übertragungsraten angehängt werden können [Cis06]. Die Antwort wird in jedem Fall als EAPOL-Paket an den Supplicant weiter geleitet.

Dieser Ablauf wurde in Abbildung 2.3 zusammengefasst.

Mittels *RADIUS-Zonen (Realms)* kann man Clients zu verschiedenen Gruppen zusammenfassen. Die Zonen werden beim Einwählen als Teil der Kennung angegeben. Die Struktur ist hierbei <*Benutzername*>@<*RADIUS-Zone*>. Über die Realm wird vor allem bestimmt, welcher Authentication Server zur Überprüfung der Login-Daten herangezogen werden soll. Außerdem können verschiedene Zugriffsberechtigungen abhängig von der Zone vergeben werden.

3 Anforderungen

In diesem Kapitel soll erst ein Überblick darüber gegeben werden, welche Technik das LRZ bereits einsetzt. Aus dieser, sowie verschiedenen Use-Cases, soll dann erarbeitet werden, welche Anforderungen ein neues Konzept zu erfüllen hat, um eine Verbesserung darzustellen.

3.1 Aktuelle IT-Infrastruktur

Das LRZ verwaltet nicht nur das eigene Netzwerk, sondern kümmert sich auch um die Anbindung der anderen Teilnehmer des Münchner Wissenschaftsnetzes an das Internet. Die größten Partizipanten sind die LMU und die TUM, deren rund 50 000 bzw 35 000 Studenten (Stand Ende 2012) automatisch eine Kennung zur Nutzung der MWN-Infrastruktur bekommen [LR13c]. Da sich eine Lösung in diese Umgebung integrieren muss, wird hier also ein Überblick über die aktuell vorhandene Infrastruktur gegeben.

3.1.1 WLAN

Um eine möglichst weite Verfügbarkeit von WLAN zu ermöglichen kamen Ende 2013 rund 2300 Access-Points vom LRZ zum Einsatz, welche über fast 300 Gebäude verteilt waren [LR13a]. Diese senden alle mindestens im 2,4 GHz-Bereich entsprechend dem 802.11g-Standard [LR13f]. Ein Großteil, nämlich etwa ein Drittel aller APs, unterstützt bereits den schnelleren 802.11n-Standard. Teilweise wird inzwischen auch schon im 5 GHz-Bereich gesendet. Es wird kalkuliert, dass pro 100 Sitzplätze eines Raumes ein Access-Point benötigt wird [LR14c]. Von diesen werden die beiden SSIDs *eduroam* und *lrz* angeboten [LR13f].

Der Unterschied zwischen den beiden ist, dass im *lrz*-Netz zwangsläufig eine VPN-Verbindung aufgebaut werden muss und deshalb meist auch eine separate Software zum Herstellen dieser benötigt wird. Ein VPN-Server, der mit einem Authentication Server in Verbindung steht, prüft zu Beginn, ob ein Nutzer die Berechtigung zur Nutzung des VPNs besitzt. Diese Berechtigung erfordert zwangsläufig eine LRZ-Kennung. Wenn diese vorhanden ist, so wird ein VPN-Tunnel zwischen Client und VPN-Server aufgebaut [LR13f]. Dies erlaubt, dass die Verbindung zwischen Client und Access-Point unverschlüsselt sein kann. Dieser Aufbau wird in Grafik 3.1 dargestellt.

Das *eduroam*-WLAN ist im Gegensatz dazu mit WPA2 gesichert und baut keinerlei VPN-Verbindung auf [LR13f]. Es hat den Vorteil, dass hier keine zusätzliche Software zum Einsatz kommen muss, sofern das Gerät den WPA2-Standard unterstützt. Eduroam ist die Ausprägung eines Projektes, welches zum Ziel hat, eine europaweite Infrastruktur zwischen Hochschulen aufzubauen, die den Mitgliedern der Teilnehmer ermöglicht, das WLAN an den jeweils anderen Hochschulen zu nutzen [Lei14]. So könnte sich beispielsweise ein Wissenschaftler einer spanischen Hochschule aus Madrid in München mit der selben Kennung anmelden, die er auch an seiner eigenen Hochschule verwendet. Realisiert wird dies über RADIUS und entsprechende RADIUS-Zonen, wie in Grafik 3.2 zu sehen ist. Dieses Projekt wird in Deutschland erweitert von der Kommission *Eduroam off Campus*, die das Ziel hat,

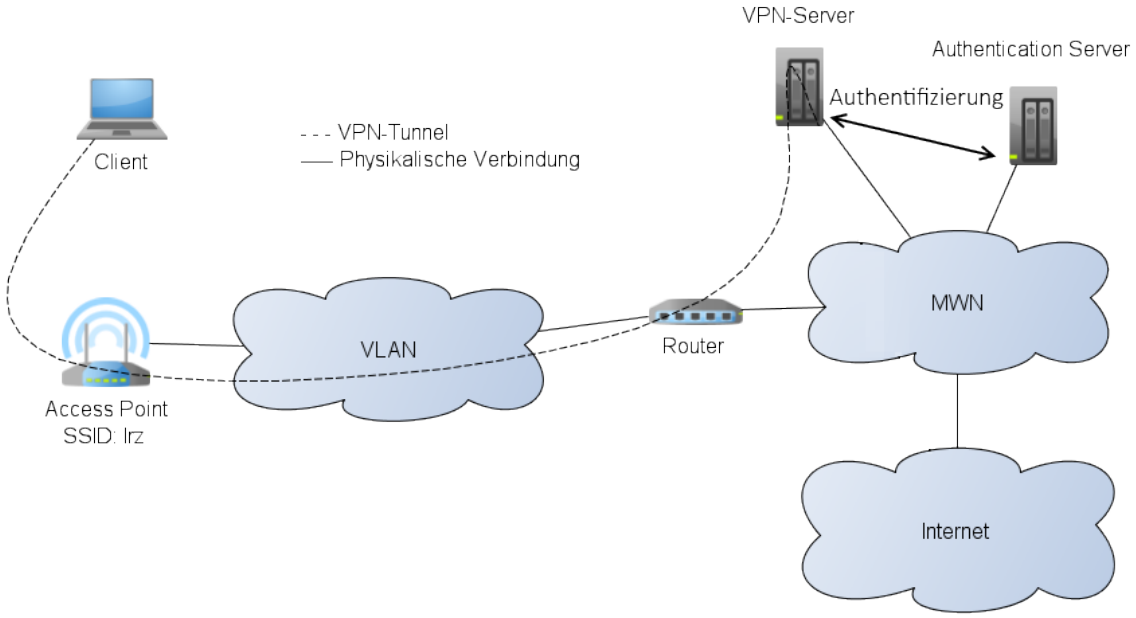

Abbildung 3.1: VPN im WLAN mit der SSID lrz

eduroam auch dort außerhalb von Hochschulflächen anzubieten, wo sich häufig Wissenschaftler und Studierende aufhalten [Zen]. Dies soll mobiles Lernen unterstützen, indem allen ein hinreichend schneller Internetzugang zur Verfügung gestellt wird. Ein gutes Beispiel für diese Erweiterung ist Ingolstadt, wo *eduroam* bereits in einem Großteil der Innenstadt verfügbar ist [bi13].

Neben diesen SSIDs wäre es denkbar, dass Institute ihre eigene SSID ausstrahlen wollen, welche auch nur von den entsprechenden Mitarbeitern verwendet werden kann. Diese müsste sich technisch nur insoweit unterscheiden, dass eine andere Datenbasis für die Benutzerauthentifizierung verwendet wird.

3.1.2 RADIUS

Damit eine andere Datenbasis für einen RADIUS-Server zur Verwendung kommen kann, soll hier zuerst ein Überblick darüber gegeben werden, wie entsprechende Anfragen bei *eduroam* verarbeitet werden.

Die RADIUS-Authentifizierungsanfragen werden an einen Server-Load-Balancer (*SLB*) weitergeleitet, an dem mehrere RADIUS-Proxy-Server - unter einer virtuellen IP-Adresse (*radius.lrz.de*) zusammengefasst - verfügbar sind [HR12]. Diese Proxy-Server sind vor allem aus Redundanzgründen mehrfach vorhanden und sollen für möglichst hohe Verfügbarkeit sorgen. Ebenso besteht der SLB eigentlich aus mehreren Servern und Rechnerpools [HR12], von denen aber immer nur ein kleiner Teil aktiv ist. Diese wurden daher nicht in Grafik 3.2 integriert.

Abhängig von der RADIUS-Zone, aus der die Anfrage kam, wird die Anfrage von diesen an unterschiedliche Ziele weitergeleitet. Das können lokale Dateien, LDAP-Verzeichnisserver, SQL-Datenbanken oder andere RADIUS-Server sein. Von diesen RADIUS-Zonen gab es im

Jahr 2012 49 Stück am LRZ, Beispiele wären *mytum.de* für TUM-Mitglieder oder *lrz.de* für die Mitarbeiter des LRZ [LR13c]. Eine vollständige Liste ist im Anhang 1 zu finden. Wenn es sich um eine internationale Anfrage handelt, die im Rahmen von *eduroam* verarbeitet wird, dann wird die Anfrage über das Internet an einen Top-Level RADIUS-Proxy weiter geleitet, der wiederum an die entsprechende Heimat-Hochschule weiterleitet [HR13b].

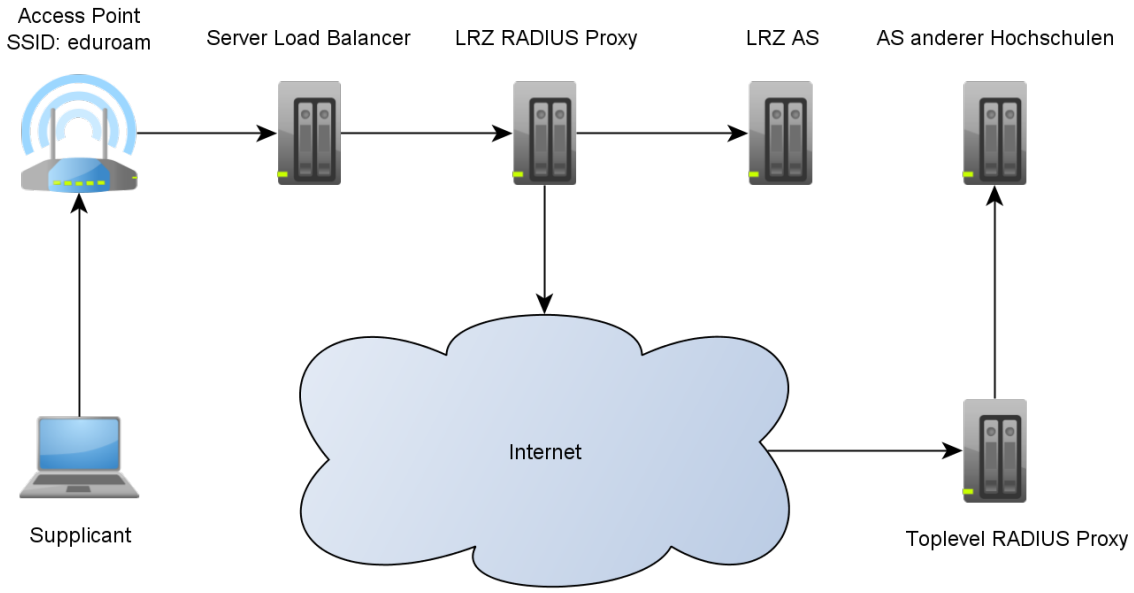

Abbildung 3.2: Beteiligte Hardware bei einer RADIUS-Anfrage im eduroam

3.1.3 Benutzerverwaltung

Um zu verstehen, wie die eben in Kapitel 3.1.2 erläuterte RADIUS-Infrastruktur Anfragen bearbeiten kann, soll hier die Funktionsweise der Benutzerverwaltung betrachtet werden. Aus diesen Informationen kann später auch erschlossen werden, wo bereits Wissen am LRZ vorhanden ist und welche Möglichkeiten aus Interoperabilitätsgründen zu bevorzugen sind.

Die Benutzerverwaltung läuft am LRZ größtenteils zentral ab. Dies hat für die beteiligten Institutionen den Vorteil, dass redundantes Abspeichern von Nutzerdaten minimiert wird und so der Speicherplatzaufwand gering gehalten wird, sowie Änderungen nicht weiterpropagiert werden müssen. Auch ein Löschen der Daten ist so einfacher. Außerdem gibt es so eine einzelne Anlaufstelle beispielsweise zum Prüfen von vergebenen Rechten, was sich in einem schnelleren Auffinden des Nutzers und damit schnellerer Abarbeitung der Überprüfung widerspiegelt. Für Nutzer hat dies auch den Vorteil, dass nur eine Kennung notwendig, und ein mehrfaches Registrieren nicht nötig ist.

Der Kernpunkt der Benutzerverwaltung am LRZ ist das *Identity-Management-System* welches aus einem Verbund von Verzeichnisdiensten besteht [LR13c]. Diese Dienste bekommen die gespeicherten Benutzerdaten primär durch Registrierung eines Benutzers beim LRZ selbst. Abgesehen davon werden sie auch automatisiert durch die jeweiligen Plattformen der LMU ($Campus^{LMU}$) sowie der TUM (*TUMonline*) gespeist [LR13c][Ebn10].

Dieses System wird primär durch LDAP-Verzeichnisdienste (auf Basis von OpenLDAP sowie Novell eDirectory) umgesetzt, welche an geeigneten Stellen durch das bereits in Windows Server integrierte *Active Directory (AD)* ergänzt werden. [LR13c][Ebn10]. ADs bieten den Vorteil, dass existierende Nutzer einfach gruppiert werden können, um deren Zugriffsrechte gemeinsam verwalten zu können.

3.1.4 Virtuelle Maschinen

Eine virtuelle Maschine *(VM)* ist ein Betriebssystem, dem von einer Software eine virtuelle Zusammenstellung von Hardware zur Verfügung gestellt wird. Dabei können mehrere dieser emulierten Betriebssysteme nebeneinander auf einer Hardwareplattform laufen. Dies erlaubt, dass Programme auf der selben Hardware in getrennten Umgebungen arbeiten können, sodass diese sich nicht untereinander beeinflussen. So können beispielsweise mehrere RADIUS-Server auf dem gleichen PC installiert werden, die zwar getrennte Datenbestände verwenden, aber ohne Veränderung von Parametern aufgespielt werden können.

Am LRZ gibt es hierfür den Dienst *Hosting virtueller Maschinen* [LR13d]. Dabei wird eine reine VM bereitgestellt, auf der ein Kunde vollen Zugriff hat und ein beliebiges Betriebssystem installieren kann. Neben diesem gibt es auch das *Managed Linux/Windows*, bei welchem sich das LRZ um Installation und Wartung des Betriebssystems kümmert, sodass der Kunde sich nur um die eigene Software kümmern muss [LR13e]. Diese Dienste werden im Analyse-Teil dieser Arbeit als eine mögliche Grundlage für einen RADIUS-Server betrachtet.

3.2 Anforderungen im Detail

Um den Instituten der Hochschulen eine einfache und effiziente Lösung anbieten zu können, wird im folgenden aufgezeigt, welche Anforderungen eine Alternative zu WLAN-Netzwerken mit PSK erfüllen muss. Dies wird anhand der verschiedenen Nutzergruppen sowie rechtlicher und technischer Einschränkungen gemacht.

3.2.1 LRZ

Rechtliche Voraussetzungen

Das Münchner Wissenschaftsnetz (*MWN*) hat einige Regeln aufgestellt, an die sich Teilnehmer halten müssen. Eine dieser Verordnungen, §2 der Richtlinien zum Betrieb des MWN, besagt, dass die Zustimmung des LRZ einzuholen ist, wenn Universitäten bzw. deren Institute eigene Hardware für den Zugang zum MWN einrichten wollen [LR14b]. Der Grund hierfür ist, dass so leichter eine einheitlichere Infrastruktur gebildet werden kann, die von vornherein ein Mindestmaß an Sicherheit erfüllt (also z.B. keinen Pre-Shared Key verwendet), sowie dass eine Kompatibilität zur bereits existierenden Technik sichergestellt werden kann. Diese Kompatibilität ist vor allem relevant für Funk-Geräte wie WLAN-Access-Points, da diese mit Geräten des LRZ interferieren könnten und so die Qualität des Netzes beeinträchtigen würden [LR14b]. Deshalb ist es auch notwendig, dass das LRZ auf Anfrage Zugang zu dieser Hardware bekommen kann [LR14b]. Die Abarbeitung der Anträge für eigene Hardware sowie eine Ausarbeitung einer individuellen Lösung kann aber viel Zeit in Anspruch nehmen, wenn es zu Interessenunterschieden kommt. Aus Sicht des LRZ wäre es also wünschenswert, wenn Institute einfach keinerlei eigene Hardware verwenden würden, die zu einer Minderung der

Netzqualität führt oder unsicher ist.

Man könnte dies realisieren, indem man die SSIDs der Institute über die bereits vorhandenen Geräte mit aussendet. Falls diese explizit einen eigenen AP wollen, so muss dieser aber so konfiguriert werden, dass keine Störung der vorhandenen Netze erfolgt und das Verbinden über 802.1X und RADIUS abgesichert ist. Als WLAN-Dienst könnte das LRZ dann anbieten, die hierfür nötigen Authentication Server auf den bereits am LRZ vorhandenen virtuellen Maschinen zu installieren und laufen zu lassen, wobei diese von den Instituten selbst verwaltet werden würden. Dieser WLAN-Dienst könnte dabei, analog zu den VM-Diensten, eine weitere Einnahmequelle für das LRZ darstellen.

Um hierbei auch noch Missbrauchsfälle bei der Verwendung des Netzes besser aufklären zu können, ist es nötig, dass eine Zuordnung von IP-Adressen zu Klarnamen möglich ist [LR14b]. Dies wird realisiert, indem einerseits der Access-Point mitprotokolliert, welche IP-Adresse er wann welcher MAC-Adresse zugeordnet hat, und andererseits der RADIUS-Server speichert, welcher Benutzername von welcher MAC-Adresse beim Login angegeben wurde. Wenn das LRZ also beide Parts verwaltet, kann sichergestellt werden, dass beide Logs vorliegen und dass diese auch schnell und kurzfristig eingesehen werden können. Einzig die Zuordnung einer IP-Adresse zu einem konkreten AP könnte sich als schwierig herausstellen, ohne welche aber nicht die korrekten Logs betrachtet werden können.

Um dieses Problem anzugreifen, kann man entweder dafür sorgen, dass die Netze der Institute nur einen eingeschränkten und damit eindeutigen IP-Adressen-Bereich bekommen. Dabei müsste zudem noch eine Übersicht angelegt werden, welches Institut welchen Bereich bekommen hat. Für eine Zuordnung einer IP-Adresse zu einem Institut müsste dann nur diese Übersicht betrachtet werden. Diese Lösung ist allerdings hinsichtlich der Größe der Bereiche unflexibel, da immer der AP umkonfiguriert und die Übersicht aktualisiert werden müsste. Eine flexiblere Möglichkeit wäre, dass die Log-Files zentral gesammelt, also z.B. von einem Script vom AP zum LRZ exportiert werden. Dann könnte man über alle Log-Dateien eine Suche nach der betroffenen IP-Adresse starten und über den Log, in welchem die Suche erfolgreich war, einen Rückschluss auf das Institut ziehen (z.B. über den Dateinamen, welcher den Namen des APs beinhalten könnte).

Lebenszyklus einer VM

Der Bestellvorgang einer VM sollte möglichst wenig Zeit in Anspruch nehmen, um nicht mehr Kapazitäten von Mitarbeitern einzunehmen, als nötig. Hierzu wäre es vorteilhaft, wenn sich das System größtenteils per Skript aufspielen ließe und insgesamt nur wenig Interaktion benötigte. Solch ein möglicher, kompletter Bestellvorgang wird im Folgenden beschrieben:

1. Der IT-Verantwortliche eines Instituts füllt ein Bestellformular auf der Webseite des LRZ aus. Um Zugriff darauf zu bekommen, muss er sich vorher mit seiner LRZ-Kennung und seinem Passwort anmelden, um sicherzustellen, dass nur Anfragen von Mitarbeitern der Hochschulen abgeschickt werden. Im Formular gibt er seinen Namen, eine e-Mail-Adresse, das vertretene Institut und optional eine individuelle RADIUS-Zone an. Außerdem muss er angeben, ob er eine eigene SSID haben möchte oder nicht und ob diese von den LRZ-Geräten ausgestrahlt werden soll. Falls dies der Fall ist,

so muss er außerdem angeben, wie diese heißen soll und in welchen Räumen diese ausgestrahlt werden soll. Als Vorlage für dieses Formular kann das Formular für die Testphase von Konferenz-WLAN herangezogen werden [LR14a].

2. Das System führt zuerst eine syntaktische Prüfung von Daten wie der e-Mail-Adresse durch. Danach wird kontrolliert, ob die gewünschte RADIUS-Zone und ggf. die SSID noch frei sind oder bereits von jemand anderem belegt oder angefragt wurden. Wenn beide Tests erfolgreich waren, werden die Eingaben akzeptiert, ansonsten wird eine entsprechende Fehlermeldung angezeigt und es wird die Chance zum Korrigieren der Eingaben gegeben. Ist alles in Ordnung, wird dem Nutzer bestätigt, dass seine Eingaben übermittelt werden. Diese werden dann in einer zentralen Anfragen-Datenbank gespeichert.

3. Ein Mitarbeiter am LRZ wird per e-Mail über die Bestellung benachrichtigt und kann dann ein Programm zur Verwaltung solcher Anfragen starten. Dieses kann entweder webbasiert oder auf dem Rechner des Mitarbeiters installiert sein. Dieses Programm müsste beim Start die Datenbank auslesen, diese auf Duplikate überprüfen, und würde dann in einer Tabelle alle Informationen der offenen Anfragen auf einen Blick anzeigen. Wenn es widersprüchliche Informationen gibt, z.B. verschiedene SSIDs von der gleichen Person für die gleichen Räume, dann werden diese farblich abgehoben dargestellt. Der Mitarbeiter soll prüfen, ob der Name der RADIUS-Zone und der SSID sinnvoll gewählt sind und welche weiteren Schritte notwendig sind: Wenn es widersprüchliche oder nicht akzeptable Anfragen gibt, dann muss mit dem Antragsteller Kontakt aufgenommen werden. Wird eine eigene SSID benötigt, so muss eine Anfrage zur Einrichtung der neuen SSID an das entsprechende Team gestellt werden, welche diese dann an den gewünschten APs konfigurieren.

4. Sind alle Informationen eindeutig und akzeptabel, so wird vom LRZ-Mitarbeiter ein *Managed Linux* (vgl. 3.1.4) angefordert, wobei mitgeteilt wird, dass dies im Rahmen des WLAN-Dienst-Angebots geschehen soll. Durch diesen Zusatz ist klar, dass im „Post-Install“ der VM automatisch das Installationsskript für den RADIUS-Server von einem anderen Rechner im MWN heruntergeladen werden soll. Ist dieser Schritt komplett, so wird das dem LRZ-Mitarbeiter über das Verwaltungsprogramm angezeigt. Entweder per Hand oder über ein Skript wird dann die Konfiguration des RADIUS-Proxys so aktualisiert, dass die gewählte RADIUS-Zone an den neuen Authentication Server weiterleitet. Hierbei muss auch ein Passwort für die Kommunikation zwischen den beiden Servern festgelegt werden, welches von einem Zufallsgenerator vorher generiert wurde. Dieses Passwort wird gecached und als Parameter beim Ausführen des Skripts zur Installation des eigentlichen Authentication Servers mit angegeben.

5. Falls es keine Probleme gab und die Installation erfolgreich abgeschlossen wurde, wird eine Bestätigungsmail an den Besteller versendet, welche die RADIUS-Zone und eine Benutzername/Passwort-Kombination enthält, die dieser benötigt, um Nutzer verwalten zu können. Ab dem Zeitpunkt der Bestätigung beginnt auch der Abrechnungszeitraum, ab welchem der Kunde zahlen muss. Der Eintrag in der Datenbank wird als abgearbeitet markiert und in eine separate Tabelle verschoben, wodurch eine Protokollfunktion erreicht wird.

3.2.2 Institute

Einrichtung

Den Instituten muss ein Produkt geboten werden, welches einfach umzusetzen und dennoch sicher ist. So sollte auf keinen Fall ein Pre-Shared Key verwendet werden, sondern statt dessen RADIUS zum Einsatz kommen. Wenn ein entsprechender Authentication-Server am LRZ stünde, müssten sich Verantwortlichen der Institute nicht in RADIUS einarbeiten, da dieses Wissen dort bereits vorhanden ist.
Außerdem muss keine zusätzliche Hardware von den Instituten angeschafft werden, auf der ein solcher Server laufen könnte. Dies erspart nicht nur die Kosten der Hardware, sondern auch die Zeit der Recherche nach passenden Geräten. Dies lässt sich analog für das LRZ erreichen, wenn statt einem echten Server eine der dort verfügbaren virtuellen Maschinen (vgl. 3.1.4) verwendet wird. Die vorteilhaftere Variante wäre hier eine *Managed VM*, da sich Institutsmitarbeiter dann um nichts zu kümmern hätten.

Damit der Bestellprozess möglichst einfach gehalten wird, könnte dieser über ein Formular auf der LRZ-Webseite abgewickelt werden. Dies funktioniert wie in 3.2.1 *Lebenszyklus einer VM* beschrieben.

Benutzer hinzufügen

Um für Flexibilität der Institute zu sorgen, sollten diese ihre benötigten Benutzerkennungen selbst verwalten können. Dies hat den Vorteil, dass keine externe Instanz den Anlegeprozess verlangsamt und führt so zu reduziertem Aufwand für alle Mitarbeiter.
Beim Anlegen der Benutzer sollten neben einem Login-Namen und Passwort aber auch weitere Daten angegeben werden, um zum Beispiel die zuvor geforderte Zuordnung zu einem Klarnamen zu realisieren. Je nach Einsatzgebiet müssen aber nicht alle Informationen notwendigerweise angegeben sein, weshalb eine Unterscheidung von verpflichteten und optionalen Daten möglich sein sollte.
Eine automatische Überprüfung auf Vollständigkeit und korrekte Syntax der erforderlichen Daten bei Eingabe wäre hierbei wünschenswert. Dies könnte über ein entsprechendes grafisches Interface realisiert werden. Neben Feldern für Benutzername/Passwort, echtem Namen sowie einer Möglichkeit zur Kontaktaufnahme (Telefonnummer; E-Mail-Adresse) wären hier optionale Felder für Arbeitsgruppen und MAC-Adressen denkbar (vgl. 3.2.2 *Zugriffskontrolle*).
Der Zugriff auf diese grafische Oberfläche könnte entweder über ein eigens für diesen Zweck entwickeltes Programm geschehen, welches auf dem Rechner des Verwaltenden ausgeführt werden muss. Dieses würde sich nach Eingabe von Login-Daten mit dem Rechner verbinden, auf dem der Authentication Server ausgeführt wird und so die Änderungen an den Benutzerdaten vornehmen. Dieses Programm müsste dann allerdings für eine Vielzahl von Betriebssystemen entwickelt werden, um eine breite Kompatibilität zu bieten und so den gesamten Prozess möglichst simpel zu halten.
Alternativ besteht die Möglichkeit, die Oberfläche als Webseite zu verwirklichen, wodurch eine Konfiguration einfach im Browser geschehen kann und so betriebssystemunabhängig funktioniert und von jedem webfähigen Gerät aus durchgeführt werden kann. Diese Webseite würde von dem Rechner bereitgestellt werden, auf dem der RADIUS-Server laufen würde. Um einen Benutzer hinzuzufügen würde man die Adresse von diesem Server im Browser

ansteuern, sich einloggen und die erforderlichen Daten eintragen.
Da manche Institute ihre Mitarbeiter bereits in Datenbanken führen, wäre es praktisch, wenn diese importiert werden könnten. Die Verwaltungsoberfläche müsste dann einen Knopf bieten, bei der die Datenbank ausgewählt werden kann. Die so importierten Nutzer sollten zur Kontrolle auf dem Bildschirm ausgegeben werden, was ein zeitgleiches Ergänzen von fehlenden, aber notwendigen Daten ermöglicht.
Letztendlich funktioniert das Sicherheitsmodell auch nur zuverlässig, wenn ausschließlich ausreichend vertrauenswürdige Personen hinzugefügt werden. Um dies leichter umzusetzen sollte nicht jeder Mitarbeiter eines Instituts die entsprechenden Rechte bekommen, sondern ein Einzelner hierfür als Verantwortlicher ausgewählt werden. Dieser wäre Ansprechpartner für Mitarbeiter und Gäste, die Zugang benötigen, aber auch Kontaktperson des LRZ, sofern Probleme auftreten.

Benutzer löschen

Wenn ein Mitarbeiter ausscheidet, muss es auch möglich sein, diesen wieder zu entfernen, um den weiteren Zugriff auf das Netz für ihn zu unterbinden. Damit dies einheitlich funktioniert und leicht zu finden ist, sollte diese Funktion über die selbe grafische Oberfläche realisiert werden, welche schon zum Hinzufügen dient.

Auf Grund einer Protokollierung von Zugriffen bezüglich der Missbrauchsaufklärung sollte dieser Lösch-Vorgang in zwei Schritten passieren. Der erste Schritt wäre nur eine Deaktivierung des Zugriffs über ein gesetztes Flag oder eine zweite, getrennte Datenbank, in welche die Benutzer verschoben werden würden und welche nicht vom RADIUS-Server betrachtet werden sollte. Ein zweiter, optionaler Schritt wäre dann, die Benutzer automatisiert zu löschen. Dies würde erst nach einer Frist von einigen Tagen geschehen, wie sie auch bei den vom LRZ angebotenen Diensten verwendet wird [LR13b]. Wenn dieser Schritt nicht obligatorisch ist, erlaubt dies, beurlaubten Mitarbeitern den Zugriff zu unterbinden, ohne dass diese bei ihrer Rückkehr komplett neu eingetragen werden müssen.
Wenn man den ersten Schritt ebenfalls automatisieren kann, hat man auch die Möglichkeit, Gästen des Instituts einen Zugang zum Internet zu verschaffen, der wieder entfernt wird, wenn der Besucher das Institut verlässt. Dies würde so funktionieren, dass es beim Eintragen des Benutzers ein Feld für ein Ablaufdatum der Kennung gibt, nach welchem diese deaktiviert wird. Dies hätte den Vorteil, dass kein Mitarbeiter sich aktiv darum kümmern müsste, und so Arbeitszeit gespart wird und dieser Schritt nicht vergessen werden kann.

Zugriffskontrolle

Damit nur Mitarbeiter auf institutsinterne Daten zugreifen können wäre eine Unterteilung von Mitarbeiteraccounts und Gastaccounts wünschenswert. Diese Trennung könnte beispielsweise durch ein VLAN-Attribut in der Access-Accept-Nachricht des Authentication Servers geschehen. Dazu müsste beim erstmaligen Anlegen eines Benutzers zusätzlich angegeben werden, ob es sich um einen Mitarbeiter handelt oder nicht. Sofern eine feinere Aufteilung gewünscht ist, wäre es auch denkbar, mehrere VLANs zu erstellen und damit die Benutzer in mehr als nur zwei Gruppen zu separieren. Der RADIUS-Server muss allerdings entsprechend konfiguriert werden, was entweder nach einer ausgehändigten Anleitung von dem Verantwortlichen des Instituts ausgeführt oder bei der Bestellung mit angegeben werden muss,

damit sich ein LRZ-Mitarbeiter darum kümmert.

Wenn gewünscht sollte es auch möglich sein, dass nur Arbeitsgeräte im Institutsnetz zugelassen werden. So wird verhindert, dass Mitarbeiter ihre privaten, oftmals weniger geschützten Geräte mit ins interne Netz bringen. Diese Ausgrenzung ließe sich über das Speichern und spätere Abgleichen der MAC-Adresse der Geräte realisieren, da diese theoretisch für jeden Netzwerkadapter eindeutig sein sollte. Um dies umzusetzen müsste sie ebenfalls mit in der Benutzerdatenbank stehen und damit beim Anlegen einer Kennung per Hand eingetragen werden. Letztendlich sollte es verschiedene Ausprägungen der Umsetzung geben, damit jedes Institut für sich entscheiden kann, wie die Geräte zugelassen werden:

- Alle Geräte sind zugelassen, eine MAC-Adresse wird nicht überprüft.
- Der Eintrag der MAC-Adresse ist optional. Nur wenn sie bei einem Nutzer hinterlegt ist, werden andere Geräte dieses Benutzers blockiert. Ist keine vorhanden, so werden alle Geräte für diese Kennung erlaubt.
- Der Eintrag der MAC-Adresse ist notwendig, ansonsten kann die Kennung nicht verwendet werden.

3.2.3 Konferenzen

Im Jahr 2012 wurde für Konferenzen und Tagungen rund 350 mal ein separates WLAN eingerichtet [LR13c]. Diese Zahl stieg im Jahr 2013 auf etwa 410 [LR13a]. Nach Vermutungen des LRZ wird diese Zahl in den kommenden Jahren noch weiter ansteigen. Im Moment laufen Konferenzen so ab, dass bei diesen eine SSID *con* erstellt wird, die jedoch unverschlüsselt ist und keinerlei Authentifizierung erfordert [Lei13]. Vorteil an dieser Konfiguration ist, dass Kongressteilnehmer keine Einstellungen an ihrem Gerät vorzunehmen haben. Die SSID wird nur für den Ort und die Uhrzeiten der Tagung aktiv geschaltet, um das Risiko von Missbrauch möglichst gering zu halten [Lei13]. Dennoch wird verlangt, dass Veranstalter mindestens vier Wochen vorher ankündigen [Lei13], ob und wann sie denn WLAN benötigen, da die Freischaltung händisch auf jedem AP des Austragungsortes erfolgen muss.
Wenn man nun auch das Konferenz-WLAN verschlüsselt und mit Authentifizierung anbieten möchte, so sollte eine Alternative deutlich flexibler und schneller einzurichten sein als ein Pre-Shared Key auf gegebenenfalls mehreren Access-Points. Bei einer Lösung via RADIUS-Server könnte eine ständige SSID ausgestrahlt und eine eigene RADIUS-Zone eingerichtet werden. Die Kennungen werden auf einem eigens für diesen Zweck vom LRZ betriebenen RADIUS-Server verwaltet. Jede Konferenz würde dann für die jeweilige Dauer eine eigene Kennung erhalten. So müssten keine Veränderungen an den Access-Points vorgenommen werden, was den Einrichtungsaufwand minimiert und damit den LRZ-Mitarbeitern Zeit spart.
Weiter optimieren ließe sich dies, indem der Output des inzwischen zu Testzwecken online gestellten Formulars zur Bestellung von Konferenz-WLAN [LR14a] automatisch so umformatiert wird, dass die zugehörige RADIUS-Nutzerdatenbank per Knopfdruck zu beschreiben wäre.
Das Problem der etwas aufwändigeren Konfiguration der Clients wird in dieser Arbeit nicht beachtet, da es im Prinzip ein spezielles Programm (*Configuration Assistant Tool, CAT* [DAN13]) gibt, welches die Einstellungen für eduroam automatisch vornimmt. Mit diesem muss nur die Kennung eingeben werden, ohne dass Rücksicht auf die genauen Verschlüsselungseinstellungen genommen werden muss.

3.2.4 Gewichtung der Anforderungen

Nicht alle Anforderungen sind gleich relevant, um eine funktionierende Verbesserung darzustellen. Deshalb wurde Tabelle 3.3 erstellt, welche den erarbeiteten Anforderungen eine Priorität zuweist, mit welcher diese Funktion umgesetzt werden sollte. Dabei bedeutet 3 unverzichtbar, 2 wichtig, 1 größerer Unterschied und 0 kaum wichtige Bonusfunktion.

Anforderung	Priorität
Einfache Verfügbarkeit einer PSK-Alternative	3
Nutzer-Selbstverwaltung der Institute	3
Logging	3
Gültigkeitsdauer von Kennungen	2
Nutzer müssen Personen zuzuordnen sein	2
Bestellung nur nach Authentifikation	2
Automatische Installation des RADIUS-Servers	1
GUI zur Nutzerverwaltung	1
Log gelöschter Nutzer	1
Bestellformular	1
Zentrale Bestellungsverwaltung	1
Zentrale Log-Sammlung	1
Benutzerimport	1
VLANs	0
MAC-Adressen-Filterung	0

Abbildung 3.3: Anforderungen an ein optimales System

4 Systementwurf

In diesem Kapitel wird ausgehend von den im vorigen Kapitel festgestellten Anforderungen ein konkreter Entwurf ausgearbeitet. Zudem wird der Auswahlprozess der benötigten Software und deren Konfiguration begründet.

4.1 Lösungsvorschlag

Um eine sichere und leicht einzurichtende Infrastruktur aufzubauen, könnte man Instituten einen vorkonfigurierten RADIUS Authentication Server samt eigener RADIUS-Zone zur Verfügung stellen. Dieser würde in einer *managed VM* (vgl. 3.1.4) auf einem Computer am LRZ laufen. Dieser Standort für den RADIUS-Server begründet sich daraus, dass er für die Institute deutlich komfortabler ist (vgl. Abschnitt 3.2.2). Für die Verwendung bei Konferenzen könnte ein zentraler, vom LRZ verwalteter RADIUS-Server zum Einsatz kommen. Bei diesem wird für jede Tagung ein eigener Benutzer samt Ablaufdatum angelegt. Dies erlaubt das ständige Ausstrahlen einer zugehörigen SSID und damit, dass niemand mehr regelmäßig Access-Points umkonfigurieren muss. Diese Arbeitszeitersparnis sorgt auch dafür, dass weniger Vorlaufzeit benötigt wird.

4.2 Auswahl der RADIUS-Software

Um eine geeignete RADIUS-Software für die VM auszuwählen, werden hier verschiedene Kriterien ausgewählt und erläutert, auf welche im Anschluss hin verschiedene Programme untersucht werden.

4.2.1 Kriterien

Eine RADIUS-Software soll nach folgenden Kriterien bewertet werden:

- Updates
  Eine der wichtigsten Voraussetzungen für eine dauerhafte Lösung ist, ob die Software in Zukunft Updates bekommt und noch weiterentwickelt wird, wodurch eventuelle Sicherheitslücken geschlossen oder bis dahin aktuellere Verfahren implementiert werden. Letztendlich zeugt eine gut gepflegte Software auch davon, dass der Entwickler Interesse an der Güte des Produkts hat.

- Support
  Für den Einsatz im LRZ, bei dem die Software sehr viel zu verwalten hat, sollte die Software möglichst wartungsarm sein. Da dies schwer vorher zu beurteilen ist, wird stattdessen betrachtet, ob der Hersteller der Software einen Wartungssupport anbietet, den das LRZ bei Problemen in Anspruch nehmen kann.

- Benutzerverwaltung
  Es sollte außerdem betrachtet werden, welche Verfahren zur Speicherung und Verwaltung von Nutzern die Programme bieten. Besonders sollte hier darauf geachtet werden, ob es nur eigene, programmspezifische Formate sind, oder ob sich auch externe Quellen oder zumindest leicht portierbare Formate darunter finden, sodass im Falle eines Programmwechsels die Datenbank übernommen werden könnte. Im Rahmen der Benutzerverwaltung sollte auch darauf geachtet werden, ob das Programm zeitlich begrenzte Gültigkeit von Kennungen unterstützt, wobei nativ verfügbar oder durch eine Erweiterung ergänzt hier gleich gewichtet werden.

- PEAP
  Die Unterstützung für PEAP *(Protected Extensible Authentication Protocol)* muss betrachtet werden, da die aktuelle Konfiguration im *eduroam* dies verwendet. Eine institutsbezogene Lösung sollte möglichst gleich gehalten werden, um die Infrastruktur übersichtlich zu halten und eine bewährte, sichere Technik zu übernehmen.

- Preis und Lizenz
  Sofern schlussendlich keine klare Entscheidung gefällt werden kann, da sich mehrere Programme im Funktionsumfang zu sehr ähneln, so kann der Preis oder die Lizenz das ausschlaggebende Kriterium darstellen.

4.2.2 Programme

Bei der Bewertung der Programme wurde eine Tabelle erstellt, welche das Vergleichen untereinander vereinfacht. Tabelle 4.1 zeigt bei den Recherchen gefundene Programme und wie diese sich in den vorher gewählten Kriterien schlagen.

Die Kandidaten wurden bei einer Onlinesuche mit der Menge der Schlüsselwörter *radius, server, software* gefunden, welche in verschiedenen Reihenfolgen und Zusammensetzungen verwendet wurden. Dabei handelt es sich um die Programme *TekRadius, RADIUS Gnu, FreeRADIUS, FreeRADIUS for Windows* sowie *Network Policy Server (NPS).*

Wenn man nun betrachtet, in welchem zeitlichen Abstand sie Updates bekommen haben, und wann das letzte Update geschehen ist, so fällt auf, dass *RADIUS Gnu* seit 2008 kein Update mehr erhalten hat. *FreeRADIUS for Windows* kann hier nur schwer eingeschätzt werden, da das Programm erst im Mai 2013 veröffentlicht wurde und im Dezember des selben Jahres auf den der zum damaligen Zeitpunkt aktuellsten Linux-Version entsprechenden Stand aktualisiert wurde. *NPS* ist Teil von Windows Server und erhält damit über Windows selbst die neuesten Updates, wobei die echte Frequenz von Aktualisierungen nur sehr schwer nachvollzogen werden konnte. Gut abgeschnitten haben hier *FreeRADIUS*, welches einem 3-monatigen Patch-Zyklus für seine stabile Version folgt, sowie *TekRADIUS*, welches über den Verlauf dieser Arbeit mehrere Updates bekommen hat.
Support für das jeweilige Produkt wird nur von *FreeRADIUS* und *TekRadius* angeboten - bei den anderen Kandidaten konnte nichts dergleichen gefunden werden.
Bezüglich der Benutzerverwaltung verhalten sich die meisten Programme ähnlich: Sie unterstützen im Regelfall alle mindestens eine der weitverbreiteten Methoden SQL, LDAP sowie Active Directory.
Eine zeitgesteuerte Kennung wird nur von *TekRadius* sowie den beiden *FreeRADIUS*-Varianten von Haus aus angeboten, bei letzteren allerdings nur für die users-Datei. *RADIUS Gnu* ist

aber programmierbar, sodass sich diese Funktionalität nachträglich einbinden lassen sollte. Bei *NPS* konnten keine Informationen gefunden werden, ob solch eine Funktionalität einzurichten ist.
Vier der fünf ausgewählten Programme unterstützen PEAP ohne Probleme, nur *RADIUS Gnu* tut dies nicht.

Die untersuchten Programme waren im Prinzip alle als Freeware verfügbar, wobei *TekRadius* zusätzlich zur kostenfreien auch eine kostenpflichtige Version anbietet. Die Vorteile dieser bieten aber für den gewünschten Einsatzzweck keinen Mehrwert und werden nicht näher betrachtet. Ein Blick auf die Lizenzen verrät, dass *FreeRADIUS* im Gegensatz zu seinen Konkurrenten Open-Source ist und unter der GNU GPLv2 entwickelt wird.

Für Linux ist *FreeRADIUS* eindeutig die bessere Wahl für einen RADIUS-Server, da es im Gegensatz zu *RADIUS Gnu* PEAP unterstützt und noch weiter entwickelt wird, es Support-Pakete zu kaufen gibt und es vor allem Open-Source ist. Bei Windows wäre *TekRADIUS* zu bevorzugen, da es explizit ausgeschriebenen Support bietet und mehr Optionen zur Abfrage von Benutzerdaten bietet als *NPS* beziehungsweise eine Konfiguration per GUI zulässt im Gegensatz zu *FreeRADIUS for Windows*.
Welcher der beiden letztendlich verwendet wird, hängt somit von der Wahl des Betriebssystems ab.

4.3 Betriebssystem der VM

4.3.1 Windows oder Linux

Da es für Linux und für Windows gleichermaßen gute RADIUS-Server gibt, müssen hier andere Kriterien als nur die Verfügbarkeit der RADIUS-Software zur Auswahl herangezogen werden.
Zum einen kann hier der Anschaffungspreis betrachtet werden. Eine Einzellizenz für *Windows 8* kostet zwischen 120 und 280 Euro [Mic14a], je nachdem, ob es sich um die Standard-Variante oder die Pro-Variante handelt. Die Lizenzen für *Windows Server 2012 R2 Datacenter Edition* mit unbegrenzt vielen virtuellen Instanzen liegen sogar bei rund 6000 USD [Mic14b]. Hier müssen zudem noch die Kosten für den Support dazu addiert werden. Linux-Betriebssysteme sind im Gegensatz dazu im Regelfall kostenlos, sofern kein Support gewünscht ist. Mit Support befindet man sich in ähnlichen Preisklassen.
Zum anderen könnte betrachtet werden, wie gut die Installation automatisierbar ist, um den Einrichtungsaufwand möglichst gering zu halten. Da bei der Windows-Plattform ein zentraler Keyserver eingerichtet werden muss, der die Produkt-Schlüssel verwaltet, lässt sich die Installation von Windows schwerer automatisieren im Vergleich zu Linux-Systemen.
Dies gilt nicht nur für die Installation des Betriebssystems selbst, sondern auch für die Installation der RADIUS-Software. Bei Windows muss manuell die Installationsdatei aufgespielt werden. Für Linux gibt es neben des Downloads der nötigen Dateien per *wget* auch die Möglichkeit, FreeRADIUS aus dem entsprechenden Software-Repository herunterzuladen und zu installieren.
Es könnte auch noch das Erscheinen von Updates in Betracht gezogen werden. Sofern eine relativ aktuelle Version gewählt wird, stellt dies bei Windows kein Problem dar. Da dies für Linux aber von der genauen Distribution abhängig ist, wird hier davon ausgegangen, dass

Name	TekRadius	RADIUS Gnu	FreeRADIUS	FreeRADIUS for Windows	Network Policy Server
Version	4.8.6	1.6.1	2.2.3	2.2.3	keine Angabe
Webseite	[Kap]	[Poz08]	[The]	[sfr13]	[Mic12]
Betriebssystem	Windows	Linux	Linux, Solaris, Mac OSX	Windows	Windows Server 2008, 2012
Updates	ja	nein	ja	keine Angabe	keine Angabe
Support	ja	nein	ja	nein	nein
PEAP	ja	nein	ja	ja	ja
Benutzer-verwaltung	Microsoft SQL, SQ-Lite, Active Directo-ry	Interne Datenbank, MySQL, PostgreS-QL	Textdatei, LDAP, MySQL, PostgreS-QL	Textdatei, LDAP, MySQL, PostgreS-QL	Active Directory
Besonderheiten	DHCP, One Time Passwords, Nati-ves Ablaufen von Kennungen, GUI zum Editieren von Nutzern, Externe Tests einbindbar	Programmierbar (Scheme, Rewrite)	DHCP, Kerberos, Proxy-Modus, Web-Administration via PHP, Programmier-bar (Perl, Python), Externe Tests ein-bindbar	DHCP, Kerberos, Proxy-Modus, Pro-grammierbar (Perl, Python), Externe Tests einbindbar	GUI zum Editieren von Nutzern, Proxy-Modus

Tabelle 4.1: Vergleich verschiedener Radius Server Software

eine mit vergleichbarer Update-Policy ausgewählt wird.
Da Linux also leichter automatisiert zu installieren ist und FreeRADIUS einfacher aufgespielt werden kann wird ein Linux als Betriebssystem für den Radius-Server gewählt.

4.3.2 Wahl der Distribution

Am LRZ kommen im Moment verschiedene Distributionen von Linux zum Einsatz, die für unterschiedliche Zwecke verwendet werden:

- *Debian* ist eine kostenfreie Distribution, die vor allem für Endanwender-Geräte eingesetzt wird, da keine Firma dahinter steht, die Support anbieten könnte [SPI13].
- *openSuse* ist ebenfalls kostenfrei, wird im Hintergrund allerdings von der Firma SUSE weiterentwickelt. Es wird dennoch auch primär für Einzelnutzer angeboten, da bei dieser Version kein Support der Firma angeboten wird [ope13].
- *Suse Linux Enterprise Server (SLES)* dagegen hat vollen Support von der Firma SUSE und wird für Geräte verwendet, bei denen so wenig Ausfälle wie möglich passieren dürfen [SUS14].

Um eine lange Laufzeit mit möglichst wenig Aufwand von Seiten des LRZ oder der Institute zu ermöglichen, bietet sich SLES am ehesten an, da sich dann die Firma SUSE um notwendige Aktualisierungen und Problemlösungen im Bezug auf das Betriebssystem kümmern würde.

4.4 Erreichbarkeit des Servers

Prinzipiell gibt es viele Möglichkeiten, wie die Kommunikation zwischen Access-Points der Institute, RADIUS-Proxy vom LRZ und den RADIUS-Servern, die am LRZ stehen und den Instituten zugeordnet sind, ablaufen kann. Es lassen sich hier drei Kernfragen herausarbeiten, auf die das Problem reduziert werden kann:

- Wird im Access-Point (AP) der RADIUS-Server oder der RADIUS-Proxy eingetragen?
- Bekommt der RADIUS-Server eine eigene IP oder einen Port am Server-Load-Balancer?
- Falls er eine IP bekommt: Ist diese fest oder variabel und wird über einen dynamischen Domain-Name-Service als fester Name repräsentiert?

Diese Fragen sollen in den folgenden Abschnitten näher betrachtet und ihre möglichen Lösungen gegeneinander abgewogen werden.

4.4.1 Proxy oder Server im AP?

Wenn ein Access-Point von einem Client kontaktiert wird und seine Zugangsdaten schickt, so müssen diese vom AP an den Authentication Server, also den richtigen RADIUS-Server, weiter geleitet werden (vgl Kap. 2.2.1), damit dieser die Daten mit jenen Werten vergleichen kann, die in der Nutzer-Datenbank gespeichert sind. Dazu muss der AP aber wissen, welcher der richtige Server ist, und wie dieser erreichbar ist. Deshalb muss im AP hinterlegt werden, wohin die Anfragen geschickt werden sollen. Hier gibt es nun zwei Möglichkeiten:

- Den Access-Points wird die Adresse des entsprechend verantwortlichen RADIUS-Servers in die Konfiguration geschrieben.
- Die Konfiguration der Access-Points enthält die Adresse des RADIUS-Proxys, der Anfragen an den zuständigen RADIUS-Server weiterleitet.

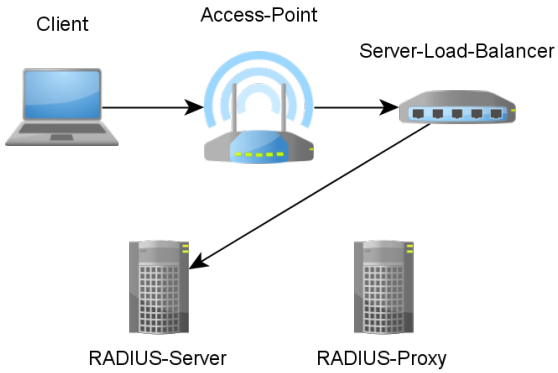

Abbildung 4.1: Möglichkeit 1 - AP schickt direkt an den RADIUS-Server

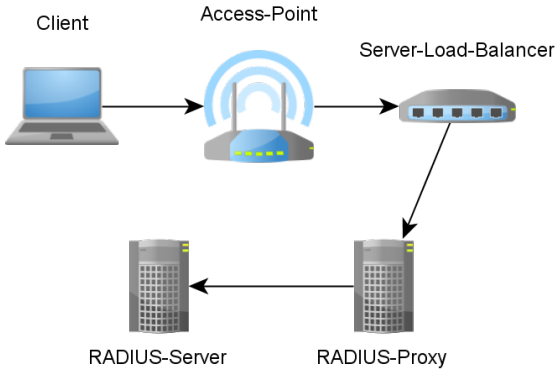

Abbildung 4.2: Möglichkeit 2 - AP schickt an den RADIUS-Proxy

Vorteil der ersten Möglichkeit ist, dass eine Anfrage schneller verarbeitet werden kann, da weniger Geräte auf dem Weg zur Nutzer-Datenbank das Paket verarbeiten müssen (vgl. Abbildung 4.1). Der Nachteil daran ist jedoch, dass sie einen erhöhten Konfigurationsaufwand beim erstmaligen Einrichten besitzt, da in jedem betroffenem Access-Point manuell die entsprechende Adresse korrekt eingetragen werden muss.
Der Nachteil von Möglichkeit 1 zeigt bereits, welchen Vorteil es hat, den RADIUS-Proxy einzuspeichern. Dieser kann einheitlich für jedes Gerät eingetragen werden und vereinfacht somit den Prozess der Auslieferung an die Institute, da nicht darauf geachtet werden muss, von wem der AP letztendlich verwendet wird. Ein weiterer Vorteil hiervon ist, dass bei einer Änderung der Adresse des RADIUS-Servers nicht jeder betroffene AP umkonfiguriert werden muss, sondern nur ein einzelner Eintrag des Proxy-Servers angepasst werden muss.

Diese Variante besitzt dafür aber den Nachteil, dass mehr Geräte bei einer Anfrage betroffen sind (vgl. Abbildung 4.2), was nicht nur zu längerer Abarbeitungsgeschwindigkeit führt, sondern auch ein höheres Risiko von ausfallender Hardware mit sich bringt.

Wenn man nun die Vor- und Nachteile der Varianten vergleicht, so besitzt die Lösung via Proxy-Eintrag mehr Vorteile. Der Nachteil der Geschwindigkeit ist im Alltag außerdem kaum zu bemerken, und das Ausfallrisiko wird durch die vorhandene Redundanz mit mehreren Proxy-Servern minimiert. Es bietet sich also an, dass die Access-Points ihre Anfragen zuerst an den RADIUS-Proxy schicken, und dieser sie dann an den korrekten RADIUS-Server weiterleitet. Der verantwortliche Server wird dabei über die mitgeschickte RADIUS-Zone ermittelt.

4.4.2 Eigene IP-Adresse oder eigener Port am SLB?

Die RADIUS-Server können entweder ganz normal eine IP-Adresse bekommen, wie für die virtuellen Maschinen im LRZ üblich. Alternativ gäbe es aber auch die Möglichkeit, dass sie einen Port am Server-Load-Balancer bekommen.

In dem Fall, dass sie am SLB angeschlossen sind, tritt mit der in Kapitel 4.4.1 erarbeiteten Lösung, dass Anfragen über den Radius-Proxy gehen, allerdings ein ungünstiger Verkehrsfluss auf. Die Anfragen würden dann vom Access-Point zum Server-Load-Balancer, weiter zum Radius-Proxy, von dort zurück zum SLB und dann weiter an den gewünschten Server gehen. Damit würde der anfallende Verkehr zwischen SLB und Proxy praktisch verdoppelt, da jedes Paket diese Verbindung zwei mal passieren muss.
Da es außerdem bisher etablierter ist, dass unterschiedliche Geräte auch über verschiedene IP-Adressen angesprochen werden, würde die Lösung über Ports am Server-Load-Balancer nur für Konfusion sorgen.
Es ist hier also günstiger, wenn man den Beispielen der bereits vorhanden RADIUS-Server folgt und diesen einfach wie üblich eine IP-Adresse zuweist, sodass Pakete direkt vom RADIUS-Proxy zum korrekten Server weitergeleitet werden können.

4.4.3 Feste oder dynamische IP-Adresse?

Für die Erreichbarkeit über eine IP-Adresse gibt es zwei Möglichkeiten: Der RADIUS-Server bekommt eine feste IP-Adresse zugewiesen, die sich nicht ändert, und unter welcher er immer zu finden ist, oder aber er hat eine dynamische IP-Adresse, welche sich ändern kann und ihm vom nächsten *Dynamic Host Configuration Protocol (DHCP-)* Server in der Geräte-Hierarchie über ihm zugewiesen wird.

Eine feste IP-Adresse ist schnell eingerichtet und besitzt außer der lokalen Eindeutigkeit und dem richtigen Subnetz keine besonderen Einschränkungen.
Die dynamische Version würde mindestens einen DHCP-Server benötigen, welcher die Verwaltung der IP-Adressen übernimmt. Damit nun aber der RADIUS-Proxy weiß, wohin die einzelnen Anfragen geschickt werden sollen, müssen die wechselnden Adressen über einen festen Namen im Rahmen eines dynamischen Domain Name Services (DynDNS) verschleiert werden. Die einzelnen RADIUS-Server müssen dann dem Service nach einem IP-Adressen-Wechsel mitteilen, wie die neue Adresse lautet.
Ein Vorteil hieran wäre, dass die Einträge im Proxy dadurch lesbarer werden könnten, da nicht mehr die IP-Adressen die Server der Institute bestimmen, sondern ein relativ frei

gewählter Name, der dann dem Namen des Instituts gleichen könnte. So könnten Veränderungen am Proxy leichter vorgenommen werden.
Da diese Veränderungen jedoch nur selten nötig sind, fällt dieser Vorteil kaum ins Gewicht. Gleichzeitig würde dies bedeuten, dass ständig ein DHCP-Server und ein DynDNS laufen und aktuell gehalten werden müssten. Man muss außerdem beim ersten Einrichten die RADIUS-Server konfigurieren, dass sie dem DynDNS die Veränderungen ihrer IP-Adresse mitteilen. Dies würde den Einrichtungsaufwand noch einmal stark erhöhen. Da FreeRADIUS aber vor allem nur beim Start die Hostnamen auflöst, und die resultierende IP-Adresse cached, müsste bei einer Änderung der IP-Adresse daraufhin auch der RADIUS-Server neugestartet werden.

Letztendlich gibt es also keinen Grund, wieso nicht eine feste IP-Adresse verwendet werden sollte, unter welcher ein RADIUS-Server dann immer zu erreichen ist.

4.5 Benutzerverwaltung

In diesem Abschnitt wird behandelt, in was für einem Format die Benutzer für den RADIUS-Server gespeichert werden. Die in den Anforderungen Kap 3.2.2 *Benutzer hinzufügen* geforderte Zuordnung zu Klarnamen sollte über eine separate Datenbank erfolgen, in der diese Klarnamen mit dem zugehörigen Benutzernamen stehen, wobei sie automatisch beim Anlegen der Nutzer über ein entsprechendes Tool aufgefüllt werden kann.

Damit der RADIUS-Server weiß, welche Benutzernamen und Passwörter erlaubt sind, muss ihm eine Liste gegeben werden, gegen welche er dies überprüfen kann. Da in Kapitel 4.2 FreeRADIUS als beste Software für diesen Entwurf beschrieben wurde, müssen die Benutzer entweder in einer Klartext-Datei gespeichert werden, ein LDAP Directory eingerichtet und verlinkt oder aber eine MySQL- oder PostgreSQL-Datenbank angelegt werden.

4.5.1 Klartext-Dateien

FreeRADIUS besitzt im Installationsverzeichnis eine Datei namens *users*, in welcher Nutzer und zu erfüllende Login-Attribute in Klartext stehen. Das Passwort kann hier entweder ebenfalls im Klartext stehen, oder aber als MD5-Hash eingetragen werden. Bei einer eingehenden Authentisierungsanfrage wird die Datei von oben nach unten hin abgearbeitet, bis der zu überprüfende Benutzername gefunden wird. Wird kein passender Eintrag gefunden, so wird die Anfrage abgelehnt. Dies passiert auch, wenn zwar der Benutzername übereinstimmt, aber das Passwort oder eines der Attribute nicht passt.
Der Vorteil an Klartext-Dateien ist, dass Benutzer sehr übersichtlich aufgeführt werden, einfach geändert werden können und keine weiteren Programme dazu installiert werden müssen.
Die Klartext-Datei hat jedoch einen gravierenden Nachteil: Der RADIUS-Server liest die users-Datei nur beim Start des Servers ein. Dies hat zur Folge, dass nach dem Hinzufügen oder Entfernen von Benutzern der RADIUS-Server immer neu gestartet werden muss, damit die neuen Informationen berücksichtigt werden können. Es gibt natürlich die Möglichkeit, dass man per Script regelmäßig überprüft, ob Änderungen aufgetreten sind, und den Server dann nur in diesem Fall neu startet. Je nach Umfang der Konfigurationsdateien kann der Einleseprozess jedoch eine gewisse Zeit in Anspruch nehmen, während welcher keine Anfragen verarbeitet werden können, sodass also niemand authentisiert werden kann und damit

neue Nutzer nicht ins Netz kommen könnten.

4.5.2 SQL oder Directory via LDAP

Um das Problem der nötigen Neustarts des Servers zu umgehen, bietet es sich an, die Benutzerdaten in ein externes System auszulagern. In diesem Fall werden die Daten nicht beim Start eingelesen, sondern die Anfragen vom RADIUS-Server an das externe System weitergeleitet. Dieses externe System kann bei FreeRADIUS entweder ein LDAP Directory oder eine MySQL- oder PostgreSQL-Datenbank sein.

Directories, die über LDAP angesprochen werden, bieten im Vergleich zu SQL einige Vorteile:

- Der Umfang und Anwendungszweck von SQL ist weiter gefasst als der von Directories, weshalb es deutlich weiter verbreitet und populärer ist. Das hat zur Folge, dass auch Angriffe auf die Datenbestände für SQL zahlreicher vorhanden sind (vgl. [MIT14a], [MIT14c] und [MIT14b]). Da dieser Entwurf eine sichere Lösung als Primärziel hat, ist dieser Nachteil von SQL von hoher Relevanz.

- Letztendlich sind die Directories vor allem auf die Nutzerverwaltung ausgelegt, weshalb viele Implementationen mit einer einfachen Möglichkeit zur Einführung von Kennwort-Richtlinien ausgestattet sind. Teilweise besitzen diese auch eine Möglichkeit, die Gültigkeit von Kennungen zu beschränken, was eine der Anforderungen aus Kapitel 3 war.

Da sich also die geforderte Eigenschaft der maximalen Gültigkeit mit Directories einfacher umsetzen lässt und diese zudem noch potentiell sicherer sind, sollten für dieses System Directories über LDAP eingesetzt werden. Zum Bearbeiten dieser Directories gibt es installierbare Editoren, die eine LDAP-Funktionalität mitbringen und zur entsprechenden IP-Adresse verbinden, sowie webbasierte Lösungen via PHP, die auf dem Server laufen würden.

4.5.3 Wahl einer LDAP-fähigen Directory-Software

Es gibt viele Anbieter, die eigene LDAP-fähige Directory-Server hergestellt haben, die sich vor allem in Zielgruppe, Funktionsumfang und Preis unterscheiden. Bewertet werden diese hier nach Updates, Support, Vorhandensein von Kennwortrichtlinien sowie der Möglichkeit, eine Gültigkeitsdauer festzulegen. Die Erweiterbarkeit des Systems durch externe Tests wird nicht betrachtet, da diese bereits durch FreeRADIUS gegeben ist und einfach von dort aus realisiert werden kann. Außerdem werden von vornherein Lösungen ausgeschlossen, die nicht auf Suse Linux Enterprise Server lauffähig sind. Die Kriterien 'Updates' und 'Support' werden aus den selben Gründen betrachtet, wie sie schon bei der Auswahl der RADIUS-Software angesehen wurden.
Eine Dauer festlegen zu können, in der einzelne Kennungen gültig sind, ist primär eine Komfortfunktion. Gleiches kann schließlich auch realisiert werden, wenn ein Datum beim User gespeichert wird, welches vom RADIUS-Server überprüft wird. Dies würde nur mehr Einrichtungsaufwand erfordern, als wenn es von Haus aus unterstützt würde. Die Kennwortrichtlinien dienen dazu, leicht sicherzustellen, dass Nutzer sichere Passwörter verwenden. Wenn die Nutzer über ein Webformular eingetragen werden, kann eine Überprüfung auch dort stattfinden. Dementsprechend zählt dies auch mehr zu den optionalen Vorteilen, das

Name	ApacheDS	Novell eDirectory	OpenLDAP	Oracle Directory Server Enterprise Edition
Version	2.0.0-M16	8.8.8	2.4.39	11.1.1.7.0
Webseite	[The14]	[Net14]	[Ope14]	[Ora13]
Updates	ja	ja	ja	zuletzt 03/2013
Support	ja	ja	nein	ja
Kennwort-richtlinien	ja	ja	ja	ja
Gültigkeit	nein	ja	nein	nein

Tabelle 4.2: Vergleich verschiedener Directory Server

Nichtvorhandensein ist aber kein Ausschlusskriterium.
Nach einer Suche im Internet nach LDAP Server Software wurde eine unvollständige Liste auf Wikipedia gefunden [Wik14a], von welchen einige prominentere zum Vergleich herausgenommen wurden. Dabei handelt es sich um ApacheDS, Novell eDirectory (welches bereits am LRZ zum Einsatz kommt), OpenLDAP sowie Oracle Directory Server Enterprise Edition (ODSEE).

Bei den meisten betrachteten Programmen gab es in letzter Zeit regelmäßig Updates. Nur bei der Software von Oracle ist das letzte Update über 1 Jahr her [Rem13], weswegen hier die Frequenz nicht verlässlich eingeschätzt werden kann. Da Novell eDirectory im Vergleich gemäß Tabelle 4.2 die meisten Komfortfunktionen von Haus aus unterstützt und bereits Erfahrung mit der Software am LRZ vorhanden ist (vgl. 3.1.3), sollte dies als LDAP-Server vor den anderen Möglichkeiten bevorzugt werden. Dies ermöglicht auch, die bereits am LRZ vorhandenen Directories zu erweitern, anstatt für jeden RADIUS-Server einen eigenen Directory-Server einzurichten. So kann nicht nur Festplattenplatz gespart werden, auch der Aufwand der Einrichtung wird minimiert, da weniger neue Verzeichnisserver installiert werden müssen.

4.5.4 Vorhandene Nutzerbestände

Manche Institute besitzen möglicherweise bereits zentrale Sammlungen ihrer Mitarbeiter, welche dann optimalerweise übernommen werden können. Dies würde das erneute Einpflegen von Nutzern unnötig machen und so dem Institutsverantwortlichen viel Zeit sparen. Außerdem müsste vom LRZ kein extra SQL-Server oder LDAP-fähiger Verzeichnisdienst auf der VM installiert werden, sondern nur der Eintrag in der Konfigurationsdatei angepasst werden, was somit auch hier Arbeitszeit sparen würde.
Damit potentiell vorhandene SQL-Datenbanken abgefragt werden können, müssen diese allerdings ein ganz bestimmtes Layout aufweisen. So werden Anfragen an die Relation *radcheck* gesendet, in welcher es die Spalten *UserName, Attribute, Value* und *Op* geben muss [Mar14]. In unserem Fall ist *Attribute* immer *„Cleartext-Password“* und *Op* ist immer „:=“, da sonst EAP nicht funktionieren würde. Sofern ein Ablaufdatum angegeben werden soll, so muss ein weiterer Eintrag mit gleichem *UserName* hinzugefügt werden, dessen *Attribute „Expiration“* heißt.

Für Verzeichnisse funktioniert die Abfrage ähnlich, sodass auch hier ein spezielles Format

gefordert ist. Zwar lassen sich die LDAP-Anfragen umkonfigurieren, damit auch andersformatierte Bestände abgefragt werden können - dies würde dem Ziel von einfacher Konfigurierbarkeit aber widersprechen. Zudem ist dem LRZ, welches die Konfiguration durchführen müsste, das Layout der an Instituten vorhandenen Bestände nicht bekannt. So würde der Kommunikationsaufwand zwischen den beiden Parteien signifikant steigen und die Zeit bis zur möglichen Nutzung würde merklich verzögert werden. Allgemein wäre also ein eigens für den RADIUS-Server konfigurierter Dienst zu bevorzugen, da der Server so nicht auf verschiedenste Daten-Schemata angepasst werden muss.

4.5.5 Fazit zur Nutzerverwaltung

Für kleine Institute mit wenigen Mitarbeitern fällt der Nachteil einer Textdatei, der Neustart des Server nach Änderungen, nicht sehr ins Gewicht. Die Wahrscheinlichkeit, dass ein Nutzer sich genau dann anmelden möchte, wenn gerade der RADIUS-Server neugestartet wird, ist nicht sehr groß, da nur sehr selten Änderungen durchgeführt werden müssen. Schlussendlich sollte den Instituten also die Wahl gelassen werden, ob sie LDAP-fähige Verzeichnisse oder Textdateien verwenden wollen. Es wäre auch denkbar, dass diese Varianten parallel eingesetzt werden.

4.6 VM-Template vs. Anleitung

Zuletzt muss man sich überlegen, ob ein vollwertiges Template für virtuelle Maschinen sinnvoll ist, oder ob nicht eine Anleitung zum eigenständigen Einrichten sinnvoller erscheint. Ein Template würde hier bedeuten, dass man ein Paket von Betriebssystem und einem RADIUS-Server erstellt, welches einfach ohne Änderung in einer virtuellen Maschine installiert werden kann.

Gegen das Template würde sprechen, dass es bezüglich Softwareupdates sehr unflexibel ist und damit in gewissen Abständen neu erstellt werden müsste. Vor allem aber stellt sich die Einschränkung, keine Änderungen an der Konfiguration mehr vornehmen zu können, als unvorteilhaft heraus, da sich die RADIUS-Server der verschiedenen Institute unterscheiden müssen. Dies tun sie unter anderem in den Accounts des verwaltenden Administrators, sowie in den Passwörtern, die für die Kommunikation zwischen RADIUS-Proxy und RADIUS-Server verwendet werden.

Eine Anleitung zum eigenständigen Einrichten für Institute würde optimalerweise nicht veralten, da einfach angegeben werden kann, dass die neuesten Versionen verwendet werden sollen. Dies funktioniert jedoch nur, wenn die neue Version sich nicht groß von der vorherigen unterscheidet. Wenn dies doch der Fall ist, so kann es sein, dass der Anleitung einfach gar nicht gefolgt werden kann. Nachdem außerdem gefordert wurde, dass sich das System sehr leicht einrichten lässt, und eine Anleitung viel Zeit zum Umsetzen in Anspruch nehmen kann und bei Problemen Hintergrundwissen erfordert, scheint dies auch nicht optimal zu sein.

Am besten wäre also eine Mischung aus beidem, die nicht nur einfach aufzusetzen, sondern auch leicht zu individualisieren ist. Dies könnte erreicht werden, indem nach der Installation des Betriebssystems ein Installationsscript für den RADIUS-Server ausgeführt wird. Dieses ist entweder interaktiv, was ständige Präsenz eines Administrators voraussetzt, oder aber es wird mit einigen Parametern gestartet, die dann für die Konfiguration eingesetzt werden.

5 Implementierung

Bei der praktischen Umsetzung des in Kapitel 4 erarbeiteten Entwurfs wurde eine funktionsfähige Installation von FreeRADIUS erstellt, die sich auf die Funktion über Textdateien beschränkt und ein Ablaufen von Kennungen unterstützt. Außerdem wurde ein Script geschrieben, welches bei Ausführung FreeRADIUS automatisch herunterlädt, installiert und für die Verwendung konfiguriert.

5.1 Aufbau der Testumgebung

Als Testumgebung für die Konfiguration wurde die kleinste am LRZ verfügbare virtuelle Maschine verwendet, da für den RADIUS-Server kaum Ressourcen benötigt werden. Als Betriebssystem kam eine 64 Bit Variante von SUSE Linux Enterprise Server 11, Patchlevel 3 zum Einsatz. Die virtuelle Maschine war dabei nur aus dem Münchner Wissenschaftsnetz erreichbar. Um die Funktionalität zu prüfen wurde über die beiden Access-Points

- ap01-0oz, Oettingenstraße 67, L001
- ap03-2wl, Boltzmannstraße 1, Institutsgebäude 2. Stock

die SSID „KleinB“ ausgestrahlt. Zudem wurde im RADIUS-Proxy eine RADIUS-Zone „KleinB“ eingetragen, sodass Anfragen an den konfigurierten RADIUS-Server gesendet werden.

5.1.1 Sicherheit

Bei EAP prüfen Clients normalerweise das Zertifikat der Gegenstelle. Bei der Installation von FreeRADIUS wird zwar ein Zertifikat generiert, aber da dieses nur selbst-signiert ist, kann die Echtheit nicht garantiert werden. Um hier die Einrichtung zu vereinfachen und dennoch eine sichere Funktionsweise zu gewährleisten, sollte der Proxy so konfiguriert werden, dass die Zertifikat-Prüfung am Proxy stattfindet und die eigentliche Anfrage vom Proxy ohne Überprüfung an den entsprechenden Server weitergeleitet wird. Die Sicherheit wird hier erreicht, indem der Server nur über das MWN erreichbar ist und die Verbindung über eine gemeinsame Passphrase geschützt ist, welche in den Konfigurationsdateien der beiden eingetragen werden muss.

Diese Trennung der Anfrage wurde realisiert, indem für die äußere, anonyme Kennung, die RADIUS-Zone KleinB.outer angegeben werden musste, welche vom Proxy nicht weitergeleitet wird. Für die eigentliche Kennung musste die Realm KleinB.inner verwendet werden, welche an den Server weitergeleitet wird.

5.2 Konfiguration der RADIUS-Software

In diesem Abschnitt soll erläutert werden, welche Schritte das Script auszuführen hatte, um eine funktionsfähige Lösung aufzubauen. Das komplette Script wurde in einzelne Blöcke

unterteilt, welche sich im Anhang 2.1 bis 2.3 finden.

5.2.1 Download der Software

SUSE bietet als Paketverwaltungstool das Programm Zypper an, welchem das von openSuse verwaltete Repository der für SLES kompilierten Version von FreeRADIUS zuerst hinzugefügt werden muss. Nachdem dies aktualisiert wurde, kann per „stiller Installation“ über den Parameter -y der FreeRADIUS-Server sowie einige Zusatztools ohne Interaktion und damit vollkommen automatisch aufgespielt werden. Dabei werden die Konfigurationen in /etc/raddb/ abgelegt und die ausführbare Datei ist als /usr/sbin/radiusd zu finden.

5.2.2 Geänderte Dateien

/etc/raddb/clients.conf

Die clients.conf dient dazu, jene Geräte festzulegen, welche Anfragen an den RADIUS-Server stellen dürfen. In unserem Fall ist dies zu Testzwecken localhost sowie der RADIUS-Proxy, welcher die von den Authenticatoren gestellten Anfragen an den korrekten RADIUS-Server weiterleiten soll. Diese Anfragen kommen von radius-out.lrz.de und werden über das in 5.1.1 erwähnte Secret gesichert. Dazu muss in der clients.conf neben der IP-Adresse also auch ein Passwort eingetragen werden, welches sich genau so auch in der Konfiguration des RADIUS-Proxys wiederfindet.

/etc/raddb/users

Bei der Benutzerverwaltung in 4.5 wurden die Möglichkeiten via LDAP-fähigen Directories sowie über eine simple Textdatei erklärt. Diese Textdatei findet sich unter dem Namen /etc/raddb/users und hat das Schema

```
Nutzername Cleartext-Password := "Passwort".
```

Da ein Ablaufen von Kennungen für z.B. Gäste und Konferenzen geplant wurde, gibt es hier die Möglichkeit, über das einem Nutzernamen zugeordnete Attribut

```
Expiration := "25 Apr 2014 23:59"
```

eine maximale Gültigkeit festzulegen.

/etc/raddb/eap.conf

Die eap.conf legt einige Parameter bezüglich der Verschlüsselung fest und wird aus Kompatibilitätsgründen einfach vom LRZ übernommen. Hier wird vor allem festgelegt, welche Art von EAP bevorzugt verwendet werden soll (default_eap_type = ttls), wo die zur Identifikation verwendeten Zertifikate liegen (certdir und cadir) und über welche virtuelle Server-Instanz die verschiedenen EAP-Arten abgearbeitet werden sollen. Diese werden im nächsten Abschnitt genauer erläutert. Für ttls und peap wird dies auf virtual_server = „inner-tunnel“ gesetzt.

/etc/raddb/sites-enabled/

Dieses Verzeichnis enthält die zwei Dateien default und inner-tunnel und definiert so virtuelle Server-Instanzen, die zur Abarbeitung von Anfragen verwendet werden. So können für verschiedene Anfrage-Arten verschiedene Schritte durchgeführt werden. Dies umfasst beispielsweise das Vorgehen beim Authentifizieren, beim Autorisieren sowie Zwischenschritte zur Manipulation der Kennung oder Zuweisung zu VLANs. Die Datei inner-tunnel wird gemäß der eap.conf für EAP-Anfragen verwendet, für alle anderen Anfragen wird die default-Datei abgearbeitet.

5.3 Ablauf des Tests

Es wurden drei verschiedene Arten von Test durchgeführt, wovon zwei über das beim RADIUS-Server beigepackte Tool „radtest“ durchgeführt wurden. Für diese existiert der Eintrag localhost in der clients.conf. Test 1 wurde aufgerufen mit dem Befehl

```
radtest testuser <passwort> 127.0.0.1:18120 0 testing12345
```

und prüfte allgemein, ob der Server erreichbar und korrekt konfiguriert war, und ob der Benutzer testuser mit dem Passwort <passwort>authentifiziert werden konnte. Dabei ist testing12345 das shared secret, welches für Anfragen verwendet werden muss.
Test 2 diente zum Testen der EAP-Funktionalität:

```
radtest -t mschap testuser <passwort> 127.0.0.1:18120 0 testing12345
```

Beide Tests konnten direkt nach Installation erfolgreich abgeschlossen werden und zeigten bei einem in der Vergangenheit liegenden Ablaufdatum, ungültigem Benutzernamen oder ungültigem Passwort, dass die Authentifizierung nicht erfolgreich vollzogen werden konnte.

Test 3 wurde durchgeführt, nachdem im RADIUS-Proxy die Weiterleitung konfiguriert wurde und die beiden Access-Points eingerichtet wurden. Hier wurde als Testgerät ein Smartphone mit Android-Betriebssystem der Version 4.4.2 verwendet. Bei diesem Test wurde festgestellt, dass keine Anfragen in FreeRADIUS ankamen, das Netzwerkinterface aber erreichten. Nachdem der Port 1812, welcher für den RADIUS-Server verwendet wird, in der installierten Firewall frei geschaltet wurde, kamen die Pakete auch in FreeRADIUS an. Hierfür musste in der Datei /etc/sysconfig/SuSEfirewall2 der Eintrag

```
FW_SERVICE_EXT_UDP="1812"
```

ergänzt werden. Diese Änderung wurde deshalb im Installationsskript ergänzt. Allerdings musste festgestellt werden, dass alle Anfragen inklusive der äußeren Kennung vom Proxy weiter geleitet wurden, und nicht wie geplant dort terminierten. Dies führte dazu, dass die Anfragen vom Authentication Server abgelehnt wurden und sich der Supplicant, also das Handy, nicht mit dem Authenticator verbinden konnte. Obwohl die Hinweise bezüglich dieser Konfiguration, die in den Konfigurationsdateien vermerkt waren, exakt befolgt wurden, ließ sich dies nicht korrigieren. In weiterführenden Arbeiten muss in Kooperation mit den für die Proxy-Konfiguration zuständigen Personen eine Problembehandlung erarbeitet werden.

5.4 Anleitung zur Einrichtung

Im folgenden Abschnitt werden die Schritte erklärt, welche zur Einrichtung durchzuführen sind.

5.4.1 Anleitung für den Proxy

Am Proxy-Server muss vom LRZ ein Eintrag vorgenommen werden, welcher dafür sorgt, dass Anfragen einer bestimmten Realm an den korrekten Authentication Server weiter geleitet werden. Dieser Eintrag muss wie folgt zur proxy.conf hinzugefügt werden:

```
home_server <Institutsname> {
type = auth
ipaddr = <IP der virtuellen Maschine>
port = 1812
secret = <Passphrase zur Sicherung der Verbindung>
require_message_authenticator = yes
}
```

Zudem muss darunter die Realm selbst konfiguriert:

```
realm <Institutsrealm> {
auth_host = <Institutsname vom home_server>
}
```

Das hier eingegebene Secret muss notiert werden, da es bei der Installation des RADIUS-Servers angegeben werden muss.

5.4.2 Anleitung für den Server

Um einen RADIUS-Server für ein Institut einzurichten, müssen dank des in 5.2 erläuterten Skripts nur sehr wenig Schritte händisch erledigt werden. Zu aller erst muss eine *managed VM* mit SLES als Betriebssystem angefordert werden. Sobald diese zur Verfügung steht, kann das Skript zur Installation und Konfiguration heruntergeladen und aus dem Account des Verantwortlichen ausgeführt werden. Hierbei muss als Parameter die Passphrase angegeben werden, die im Proxy hinterlegt wurde. Wenn die Installation abgeschlossen wurde, sollte das Skript zur Benutzerverwaltung (vgl. 5.4.3) in das Home-Verzeichnis herunter geladen werden, damit es einfacher aufzurufen ist. Zuletzt können dem Verantwortlichen des Instituts die Zugangsdaten mitgeteilt werden, mit welchen er sich via SSH zur VM verbinden kann, damit er die Nutzer verwalten kann.

5.4.3 Anleitung für den Instituts-Verantwortlichen

Da im Moment kein Programm mit grafischer Oberfläche zur Verwaltung der Nutzer verfügbar ist und nicht mehr realisiert werden konnte, wurde zu diesem Zweck ein Kommandozeilentool geschrieben, welches das einfache Anzeigen, Hinzufügen und Löschen von Benutzern erlaubt, sodass Fehler in der users-Datei vermieden werden können. Der Code für dieses Tool findet sich in Anhang 3.

Dieses Programm hat 2 verschiedene Modi:

- Über die Angabe von Parametern lassen sich Aktionen durchführen, sodass keine Interaktion notwendig ist. Dieser Modus dient vor allem dazu, dass externe Programme oder Webseiten das Script ausführen können, um die users-Datei zu bearbeiten.
- Der andere Modus ist vollkommen interaktiv und wird gestartet, wenn keine Parameter angegeben werden. Hierbei werden dem Nutzer verschiedene Fragen gestellt, um die Informationen zu bekommen, welche benötigt werden.

Als Parameter für Modus 1 werden akzeptiert:

- -add Username Password (Ablaufdatum im Format 31.12.2014)
  Fügt den Benutzer mit dem Namen Username und dem Passwort Password hinzu. Das Ablaufdatum ist optional.
- -del Username
  Löscht nach Rückfrage den Benutzer Username. Damit darüber Protokoll geführt wird, wird dieser in eine Datei „delusers“ geschrieben.
- -list
  Zeigt den Inhalt der users-Datei an.

Ist die Parameter-Anzahl größer als 0, aber es wurde keiner der obigen Parameter erkannt, so werden die verschiedenen Möglichkeiten aufgelistet.

Analog hierzu wird im interaktiven Modus gefragt, ob ein Benutzer hinzugefügt oder gelöscht werden soll oder ob die Datei ausgegeben werden soll. Sofern die Option des Hinzufügens gewählt wird, erscheinen Fragen nach Benutzername und Passwort, sowie die Frage, ob bzw. was für ein Ablaufdatum verwendet werden soll. Wenn die Option des Löschens gewählt wird, dann wird nur nach dem Benutzernamen gefragt.

6 Zusammenfassung und Ausblick

Im Rahmen der Arbeit wurde die Problemstellung erläutert, dass viele Institute ihre WLAN-Zugangspunkte nur mit der unzureichenden Methode eines Pre-Shared Keys absichern. Um zu begründen, wieso dies nicht ausreicht, wurde ein Überblick über die Funktionsweise von WPA2 mit PSK gegeben, welche daraufhin in Kontrast gesetzt wurde zur besseren Alternative über 802.1X mithilfe eines RADIUS-Servers.
In einer Anforderungsanalyse wurde daraufhin beschrieben, in welche Infrastruktur sich eine Lösung einzupassen hätte. Außerdem wurden die Anforderungen der verschiedenen Nutzergruppen LRZ, Institute und Konferenzen in verschiedenen Situationen erarbeitet.
Anhand dieser Analyse wurde ein Systementwurf erarbeitet, welcher im Laufe des Kapitels immer genauer ausgearbeitet wurde. Dies umfasste nicht nur die Begründung für *FreeRADIUS* als Software und Linux als Betriebssystem, sondern auch die Konfiguration des Servers sowie der Radius-Software. Hierbei wurde auch festgestellt, dass weder ein Template für eine virtuelle Maschine noch eine Anleitung zum Selbst-Erstellen die sinnvollste Lösung sind.
In der Implementation wurde deshalb ein Skript zur automatischen Installation und Konfiguration von *FreeRADIUS* erstellt, welches nur das Passwort zur sicheren Kommunikation mit dem RADIUS-Proxy als Eingabe benötigt. Zuletzt wurde auch ein Skript zur Verwaltung von Nutzern geschrieben, um das Fehlen einer grafischen Oberfläche zu kompensieren.

6.1 Erreichte Ziele

Die Ziele der Arbeit aus Kapitel 1 konnten alle erfüllt werden. Im Rahmen dessen konnten auch die wichtigsten Anforderungen, die sich in Kapitel 3 ergeben haben, erfolgreich umgesetzt werden. Einen Überblick hierüber gibt die Tabelle in Abbildung 6.1. Vor allem die Basis, also eine funktionierende, sehr einfache Installation eines RADIUS-Servers, die von den Instituten selbst verwaltet werden kann, ist gegeben. Nicht erfüllt werden konnten primär weitergehende Dienste, die um den Server herum angesiedelt sind. Das bezieht sich beispielsweise auf den Bestellvorgang des Servers, den Import vorhandener Nutzerdatenbestände und eine zentrale Log-Verwaltung. Die gewünschte grafische Oberfläche zur Verwaltung der Nutzer ist nicht komplett gegeben, da immernoch eine SSH-Verbindung aufgebaut werden muss. Das Eintragen und Löschen konnte aber über ein Skript benutzerfreundlich gestaltet werden und soll so auch Konfigurationsfehler vermeiden.

6.2 Ausstehende Aufgaben

Die Implementation hat sich nur auf die Umsetzung über die users-Datei beschränkt. Für größere Institute sollte hier also noch die Funktionsweise über LDAP weiter erläutert und ausgearbeitet werden. Es gilt zum Beispiel noch zu klären, ob den Instituten ein Teil-Ast in den bereits am LRZ vorhandenen LDAP-Diensten zugewiesen werden kann, oder ob neben

Anforderung	Priorität	erfüllt?
Einfache Verfügbarkeit einer PSK-Alternative	3	1
Nutzer-Selbstverwaltung der Institute	3	1
Logging	3	1
Gültigkeitsdauer von Kennungen	2	1
Nutzer müssen Personen zuzuordnen sein	2	0
Bestellung nur nach Authentifikation	2	0
Automatische Installation des RADIUS-Servers	1	1
GUI zur Nutzerverwaltung	1	0,5
Log gelöschter Nutzer	1	1
Bestellformular	1	0
Zentrale Bestellungsverwaltung	1	0
Zentrale Log-Sammlung	1	0
Benutzerimport	1	0
VLANs	0	0
MAC-Adressen-Filterung	0	0

Abbildung 6.1: Anforderungen an ein optimales System

jedem RADIUS-Server auch noch ein Verzeichnisdienst installiert werden muss. Dementsprechend müsste das Skript zur Installation noch angepasst werden.
Sofern die korrekte Weiterleitung von ausschließlich der internen Kennung am RADIUS-Proxy (vgl. 5.3) doch nicht konfiguriert werden kann, so müsste man hier auch überlegen, ob es möglich ist, dass jeder RADIUS-Server automatisiert ein korrekt vom LRZ signiertes Zertifikat bekommen kann, oder wie die Kommunikation anderweitig sicher ablaufen kann.

Für die Zukunft lässt sich das System auch noch hinsichtlich einiger offener Anforderungen ergänzen. Hier wäre vor allem der Bestellvorgang interessant umzusetzen, indem das entsprechende Bestellformular erstellt und für das MWN freigeschaltet wird. Auch eine echte grafische Oberfläche, welche den SSH-Verbindungsaufbau unnötig macht, würde den Komfort für Institute erhöhen, und damit den Anreiz zum Kauf steigern. Letztendlich sollte noch erarbeitet werden, wie eine Geräteauthentifizierung über eine MAC-Adresse funktionieren würde, und wie die Zuordnung in verschiedene VLANs umsetzbar ist.

6.3 Update-Policy

Durch die Wahl eines Betriebssystems, dessen Hersteller und Vertreiber SUSE über einen Supportvertrag die Aktualität und Sicherheit gewährleistet, ist keine weitere Maßnahme bezüglich Updates in dieser Richtung notwendig.
Bei der Software FreeRADIUS gestaltet sich dies etwas komplexer, da keine Firma expli-

zit die Pflege der installierten Versionen des Programms versprochen hat. So müsste noch überlegt werden, wann, wie und welche Updates aufgespielt werden. Da die Installation im Moment aber über ein Software-Repository realisiert wird, werden auch Updates über dieses bereit gestellt. Vorteil daran ist, dass diese im Regelfall weniger Probleme enthalten, da sie erst länger getestet werden, bevor sie über das Repository verteilt werden. Dies bedeutet aber auch, dass sich dort fast nie die neuesten Versionen des Programms befinden, sondern meist ältere, welche möglicherweise Bugfixes und Sicherheitsupdates der späteren noch nicht enthalten. Sofern sich bei der Verwendung zeigt, dass die Flexibilität von selbst kompilierten Versionen abgeht, so müsste dies über entsprechende Änderungen im Installationsscript realisiert werden.

6.4 Anwendbarkeit bei anderen Hochschulen

Die erarbeitete Lösung ist von ihrer Konfiguration bezüglich z.B. Verschlüsselungen oder der Kommunikation mit einem RADIUS-Proxy sehr an die bereits im Münchner Wissenschaftsnetz vorhandene Hard- und Software angepasst. Über eine Anpassung dieser Daten im Installationsscript kann aber eine Kompatibilität mit anderen Hochschulen erreicht werden, sodass auch diese ihren Instituten eine sicherere Alternative zu einer Authentisierung via Pre-Shared Key anbieten könnten.

Abbildungsverzeichnis

Literaturverzeichnis

[bi13] INGOLSTADT BLICKPUNKT: *Eduroam off Campus in der Ingolstädter Innenstadt*, 10 2013. http://blickpunkt-ingolstadt.de/meldungen/9673-eduroam-off-campus-in-der-ingolst%C3%A4dter-innenstadt.html.

[Cis06] CISCO: *How Does RADIUS Work?*, 01 2006. http://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/12433-32.html.

[Cis12] CISCO: *802.11 Sniffer Capture Analysis - WPA/WPA2 with PSK or EAP*, 05 2012. https://supportforums.cisco.com/docs/DOC-24494.

[Com13] COMSCORE MOBILENS: *Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2013 (in Millionen)*, 09 2013. http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/.

[DAN13] DANTE LTD.: *eduroam Configuration Assistant Tool*, 2013. https://cat.eduroam.org/.

[Ebn10] EBNER, DR. RALF: *Identity Management im Münchner Wissenschaftsnetz*, 12 2010. https://www.lrz.de/services/termine/vr-it-betrieb/ebner_2010-12-09.pdf.

[Ele14] ELEKTRONIK-KOMPENDIUM.DE: *IEEE 802.11i - WPA/WPA2 - WiFi Protected Access*, 2014. http://www.elektronik-kompendium.de/sites/net/0907111.htm.

[GMX] GMX.NET: *GMX Sicherheit im Mailprogramm.* https://hilfe.gmx.net/sicherheit/ssl.html.

[Goo14] GOOGLE: *Warum kann ich nicht mit HTTP auf Gmail zugreifen?*, 2014. https://support.google.com/mail/answer/74765?hl=de.

[HR12] HOMMEL, WOLFANG und HELMUT REISER: *Das Münchner Wissenschaftsnetz - Konzepte, Dienste, Infrastrukturen, Management*, 2012. https://www.lrz.de/services/netz/mwn-netzkonzept/mwn-netzkonzept.pdf.

[HR13a] HOMMEL, WOLFANG und HELMUT REISER: *Kapitel 10: Netzsicherheit - WLAN-Sicherheit*, 01 2013. http://www.nm.ifi.lmu.de/teaching/Vorlesungen/2012ws/itsec/_skript/itsec-k10-v8.0.pdf.

[HR13b] HOMMEL, WOLFANG und HELMUT REISER: *Kapitel 9: Netzsicherheit - Data Link Layer*, 01 2013. http://www.nm.ifi.lmu.de/teaching/Vorlesungen/2012ws/itsec/_skript/itsec-k9-v8.0.pdf.

[IEE10] IEEE: *IEEE Std 801.1X : Port-Based Network Access Control*, 02 2010. https://standards.ieee.org/getieee802/download/802.1X-2010.pdf.

[IEE12] IEEE: *Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications*, 03 2012. https://standards.ieee.org/getieee802/download/802.11-2012.pdf.

[Ins13] Institut für Demoskopie Allensbach: *Computerbesitzer in Deutschland nach Art bzw. Bauform der im Haushalt vorhandenen Computer*, 10 2013. http://de.statista.com/statistik/daten/studie/168798/umfrage/computerbesitz-in-haushalten-nach-computerart/.

[Kap] KaplanSoft: *TekRADIUS*. http://www.kaplansoft.com/tekradius/.

[Lei13] Leibniz-Rechenzentrum: *Netzkonfiguration für Veranstaltungen (Tagungen, Kongresse usw.)*, 06 2013. https://www.lrz.de/services/netz/mobil/kongress/.

[Lei14] Leibniz-Rechenzentrum: *eduroam*, 04 2014. https://www.lrz.de/services/netz/mobil/eduroam/.

[LR13a] Leibniz-Rechenzentrum: *Beitrag zum Jahrbuch 2013, Kommission für Informatik*, 12 2013. https://www.lrz.de/wir/berichte/jbkomm/jahrbuch_2013.pdf.

[LR13b] Leibniz-Rechenzentrum: *Beschränkungen und Monitoring im Münchner Wissenschaftsnetz*, 06 2013. https://www.lrz.de/services/netz/einschraenkung/.

[LR13c] Leibniz-Rechenzentrum: *Jahresbericht 2012*, 07 2013. https://www.lrz.de/wir/berichte/JB/JBer2012.pdf.

[LR13d] Leibniz-Rechenzentrum: *Leistungsbeschreibung: Hosting virtueller Server*, 05 2013. https://www.lrz.de/services/serverbetrieb/hosting_virtueller_server/.

[LR13e] Leibniz-Rechenzentrum: *Leistungsbeschreibung: Managed Linux/Windows*, 05 2013. https://www.lrz.de/services/serverbetrieb/managed_os/.

[LR13f] Leibniz-Rechenzentrum: *WLAN (Wireless LAN) im MWN*, 11 2013. https://www.lrz.de/services/netz/mobil/wireless/.

[LR14a] Leibniz-Rechenzentrum: *Anmeldung Konferenz-WLAN*, 03 2014. https://www.lrz.de/services/netz/mobil/mwn-events/.

[LR14b] Leibniz-Rechenzentrum: *Regeln für den Betrieb von Institutseigenen WLANs*, 03 2014. https://www.lrz.de/services/netz/mobil/inst-funklans/.

[LR14c] Leibniz-Rechenzentrum: *Richtlinie für die Anzahl und Platzierung von Accesspoins Öffentlichen Räumen*, 02 2014. https://www.lrz.de/services/netz/mobil/anzahl-aps/.

[Mar14] Marco ...: *guide/SQL Howto*, 02 2014. http://wiki.freeradius.org/guide/SQL-HOWTO.

[Mic12] Microsoft: *Network Policy Server*, 03 2012. http://technet.microsoft.com/library/cc732912.aspx.

[Mic14a] Microsoft: *Windows kaufen - Microsoft Store Deutschland*, 2014. http://www.microsoftstore.com/store?Action=cat&Locale=de_DE&SiteID=msde&Windows=&categoryID=64873800.

[Mic14b] Microsoft: *Windows Server 2012 R2*, 2014. https://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/default.aspx#fbid=T0PS3tZzFyj.

[MIT14a] MITRE Corporation: *Mysql Vulnerability Statistics*, 04 2014. http://www.cvedetails.com/vendor/185/Mysql.html.

[MIT14b] MITRE Corporation: *Openldap Vulnerability Statistics*, 04 2014. http://www.cvedetails.com/vendor/439/Openldap.html.

[MIT14c] MITRE Corporation: *Postgresql Vulnerability Statistics*, 04 2014. http://www.cvedetails.com/vendor/336/Postgresql.html.

[Net14] NetIQ Corporation: *Full-service, secure directory*, 2014. https://www.netiq.com/products/edirectory/.

[ope13] *openSuse Wiki Hauptseite*, 11 2013. https://de.opensuse.org/Hauptseite.

[Ope14] OpenLDAP Foundation: *OpenLDAP Download*, 2014. http://www.openldap.org.

[Ora13] Oracle Corporation: *Oracle Fusion Middleware Evaluation Guide for ODSEE 11gR1*, 2013. http://docs.oracle.com/cd/E29127_01/doc.111170/e28968/toc.htm.

[Por13] Portal München Betriebs-GmbH & Co. KG: *Fragen und Antworten zum Surfen mit M-WLAN*, 2013. http://www.muenchen.de/leben/wlan-hotspot/anleitung.html.

[Poz08] Poznyakoff, Sergey: *GNU radius - Zusammenfassung*, 2008. https://savannah.gnu.org/projects/radius/.

[Rem13] Remillon, Etienne: *ODSEE 11gR1 PS2 Released*, 03 2013. https://blogs.oracle.com/directoryservices/entry/oracle_directory_server_enterprise_edition1.

[rfc04] *Extensible Authentication Protocol (EAP)*, 06 2004. http://tools.ietf.org/html/rfc3748.

[sfr13] sfreschi: *FreeRADIUS for Windows*, 12 2013. http://sourceforge.net/projects/freeradius/.

[SPI13] SPI: *Getting Debian*, 12 2013. https://www.debian.org/distrib/.

[SUS14] SUSE: *How to Buy SUSE Linux Enterprise Server Subscriptions*, 2014. https://www.suse.com/products/server/how-to-buy/.

[The] The FreeRadius Team: *Features of the FreeRADIUS AAA Server*. http://freeradius.org/features.html.

[The14] The Apache Software Foundation: *ApacheDS - LDAP and Kerberos server written in Java*, 03 2014. https://directory.apache.org/apacheds/.

[Umf12] Umfragezentrum Bonn: *Arbeitshetze, Arbeitsintensivierung, Entgrenzung - So beurteilen die Beschäftigten die Lage*, 03 2012. http://www.dgb-index-gute-arbeit.de/downloads/publikationen/data/arbeitshetze_arbeitsintensivierung_entgrenzung_-_ergebnisse_der_repraesentativumfrage_2011.pdf.

[Wik14a] *List of LDAP Software - Server Software*, 02 2014. https://en.wikipedia.org/wiki/List_of_LDAP_software#Server_software.

[Wik14b] *Peer authentication methods*, 01 2014. https://en.wikipedia.org/wiki/IEEE_802.11s#Peer_authentication_methods.

[Zen] Zentren für Kommunikation und Informationsverarbeitung e.V.: *Kommission Eduroam off Campus*. https://www.zki.de/arbeitskreise/kommission-eduroam-off-campus/.

Anhang

1 RADIUS-Zonen am LRZ

Radius-Zone	Institut	Radius-Zone	Institut
LRZ-Kennung (ohne Zone)	LRZ	lmu.de	Internet und Virtuelle Hochschule (LMU)
studlmu	LRZ	lpr.tum	Lehrstuhl für Prozessrechner
studext	LRZ	lrz.de	LRZ-Mitarbeiter
aci.ch.tum	Lehrstuhl für Anorganische Chemie TUM	nm.informatik.lmu	Institut für Informatik der LMU
binfo.wzw.tum.de	Genome oriented Bioinformatics	nmtest.informatik.lmu	Institut für Informatik der LMU
bmo.lmu	LS Biomolekulare Optik LMU	math.lmu.de	Mathematisches Institut LMU
campus.lmu.de	Internet und Virtuelle Hochschule (LMU)	math.tum	Zentrum Mathematik TU-München
cicum.lmu	Department Chemie LMU	med.lmu	Medizin der LMU, Großhadern
cip.informatik.lmu	Institut für Informatik der LMU	med.lmu.de	Medizin der LMU, Großhadern
cipmath.lmu	Mathematisches Institut LMU	meteo.lmu	Meteorologisches Institut LMU
edv.agrar.tum	Informationstechnologie Weihenstephan (ITW)	mytum.de	TUM
fh-weihenstephan.de	Fachhochschule Weihenstephan	physik.lmu.de	Physik LMU
fhm.de	Hochschule München	radius.wzw.tum	Informationstechnologie Weihenstephan (ITW)
fhm.edu	Hochschule München	rcs.tum	Lehrstuhl für Realzeit-Computersysteme
forst.tum	Forstwissenschaftliche Fakultät	rz.fhm	Rechenzentrum der Hochschule München (Studenten)
frm2.tum	Forschungsreaktor	sec.in.tum.de	Chair for IT Security, TUM
fsei.tum	Fachschaft Elektro- & Informationstechnik	sec.in.tum.gaeste	Chair for IT Security, TUM
fsmpi.tum	Fachschaften MPI	staff.fhm	Rechenzentrum der Hochschule München (Mitarbeiter)
hm.edu	Rechenzentrum der Hochschule München	tum.de	TUM
hswt.de	Hochschule Weihenstephan-Triesdorf	usm	Uni Sternwarte
ibe.lmu	Institut für medizinische Informationsverarbeitung und Biometrie und Epidemiologie	usm.lmu	Uni Sternwarte
ikom.tum	Fachschaft Elektro- & Informationstechnik	vm08.fhm	Fachbereich 08, HM
info.tum	Informatik TUM	wzw.tum	Informationstechnologie Weihenstephan
lkn.tum	Lehrstuhl für Kommunikationsnetze		

2 Script zur Ersteinrichtung

2.1 Prolog

```
#!/bin/bash
###   Parameter 1 wird als Passwort uebernommen ###
###   Wenn kein Parameter vorhanden, frage nach   ###
###   einem und beende das Script         ###
if [ -z "$1" ]; then
  echo "Bitte das Secret mit dem Proxy als Parameter angeben"
  exit;
else
```

2.2 Installation der Software und Konfiguration der Firewall

```
  ###   Installation der Software         ###
  sudo zypper ar http://download.opensuse.org/repositories/
     network:/aaa/SLE_11/ FreeRadius
  sudo zypper ref FreeRadius
  sudo zypper in -y freeradius-server freeradius-client
     freeradius-server-utils
  ###   Setzen der noetigen Berechtigungen      ###
  echo "Installation complete, updating fileownership"
  export user=$(whoami)
  sudo chown -R "$user" /etc/raddb
  sudo chown -R "$user" /usr/share/freeradius
  sudo chown -R "$user" /var/log/radius
  sudo chown -R "$user" /var/run/radiusd
  sudo chown "$user" /usr/bin/radtest
  sudo chown "$user" /usr/sbin/radiusd
  ###   Konfiguration der Firewall          ###
  sudo sed "s/FW_SERVICES_EXT_UDP=\"\"/FW_SERVICES_EXT_UDP=\"
     1812\"/g" /etc/sysconfig/SuSEfirewall2 > SuSEfirewall123
  sudo mv /etc/sysconfig/SuSEfirewall2 /etc/sysconfig/
     SuSEfirewall_prerad
  sudo mv SuSEfirewall123 /etc/sysconfig/SuSEfirewall2
```

2.3 Konfiguration

```
  ###   Konfiguration des Servers         ###
  echo "Updating config-files"
  echo "client localhost {
      ipaddr  = 127.0.0.1
      secret  = testing12345
      require_message_authenticator = no
      nastype = other
  }

```

```
10  client radius.lrz.de {
11          ipaddr = 129.187.254.16
12          secret = $1
13          require_message_authenticator = no
14  }"> /etc/raddb/clients.conf
15  echo 'testuser        Cleartext-Password := "radtestpassw0rd"
       , Expiration := "25 Apr 2014"' > /etc/raddb/users
16  echo 'eap {
17    default_eap_type = ttls
18    timer_expire = 60
19    ignore_unknown_eap_types = no
20    cisco_accounting_username_bug = no
21    max_sessions = 2048
22  # Supported EAP-types
23    tls {
24      certdir = ${confdir}/certs
25      cadir = ${confdir}/certs
26      private_key_password = whatever
27      private_key_file = ${certdir}/server.pem
28      certificate_file = ${certdir}/server.pem
29      dh_file = ${certdir}/dh
30      random_file = ${certdir}/random
31      cipher_list = "DEFAULT"
32      cache {
33        enable = no
34        max_entries = 255
35      }
36    }
37
38    ttls{
39      default_eap_type = md5
40      copy_request_to_tunnel = no
41      use_tunneled_reply = yes
42      virtual_server = "inner-tunnel"
43    }
44
45    peap{
46      default_eap_type = mschapv2
47      copy_request_to_tunnel = no
48      use_tunneled_reply = no
49      virtual_server = "inner-tunnel"
50    }
51
52    mschapv2{
53    }
54  }' > /etc/raddb/eap.conf
55  echo 'authorize{
```

```
56      preprocess
57      chap
58      mschap
59      suffix
60      ntdomain
61      eap{
62        ok = return
63      }
64      files
65      expiration
66      logintime
67      pap
68    }
69    authenticate{
70      Auth-Type PAP{
71        pap
72      }
73      Auth-Type CHAP {
74        chap
75      }
76      Auth-Type MS-CHAP {
77        mschap
78      }
79      eap
80    }
81    preacct{
82      preprocess
83      acct_unique
84      suffix
85      files
86    }
87    accounting{
88      detail
89      unix
90      radutmp
91      attr_filter.accounting_response
92    }
93    session{
94      radutmp
95    }
96    post-auth{
97      Post-Auth-Type REJECT {
98        attr_filter.access_reject
99      }
100   }
101   pre-proxy{
102   }
```

```
103   post-proxy{
104     eap
105   }' > /etc/raddb/sites-available/default
106   echo 'server inner-tunnel{
107     listen{
108       ipaddr = 127.0.0.1
109       port = 18120
110       type = auth
111     }
112
113     authorize{
114       chap
115       mschap
116       suffix
117       eap {
118         ok = return
119       }
120       files
121       expiration
122       logintime
123       pap
124     }
125
126     authenticate {
127       Auth-Type PAP {
128         pap
129       }
130       Auth-Type CHAP {
131         chap
132       }
133       Auth-Type MS-CHAP {
134         mschap
135       }
136     eap
137     }
138     session{
139       radutmp
140     }
141     post-auth{
142       Post-Auth-Type REJECT {
143         attr_filter.access_reject
144       }
145     }
146     pre-proxy{
147     }
148     post-proxy{
149       eap
```

```
150     }
151   }' > /etc/raddb/sites-available/inner-tunnel
152   echo 'expiration {
153     reply-message = "Your account has expired, %{User-Name}\n"
154   }' > /etc/raddb/modules/expiration
155   echo "Fertig konfiguriert und installiert!"
156 fi
```

3 Script zur Verwaltung der Nutzer

```
1  #!/bin/bash
2
3  ### Funktionen die benoetigt werden.  ###
4  ### Bieten Angriffspunkt fuer externe Programme ###
5  add () {
6    if [[ -n $(cat /etc/raddb/users | grep "^$1 ") ]]; then
7      echo "Nutzer $1 existiert bereits."
8      exit;
9    fi
10   if [[ "$#" == 2 ]]; then
11     echo "$1  Cleartext-Password := \"$2\"" >> /etc/raddb/users
12     echo "Nutzer $1 mit Passwort $2 wurde hinzugefuegt"
13   else
14     echo "$1  Cleartext-Password := \"$2\", Expiration := \"$3\
         "" >> /etc/raddb/users
15     echo "Nutzer $1 mit Passwort $2 und Ablaufdatum $3 wurde
         hinzugefuegt"
16   fi
17 }
18
19 checkdate () {
20   if [[ $1 =~ [0-9]{2}\.[0-9]{2}\.[0-9]{4} ]]; then
21     tag=$(echo "$1" | awk -F \. {'print $1'})
22     monat=$(echo "$1" | awk -F \. {'print $2'})
23     jahr=$(echo "$1" | awk -F \. {'print $3'})
24   else
25     echo
26     exit;
27   fi
28   if [[ "$tag" -gt 31 ]]; then
29     echo
30     exit;
31   fi
32   if [[ "$monat" -gt 12 ]]; then
33     echo
34     exit;
35   fi
```

```
36    monat2=" "
37    case "$monat" in
38      01)
39      monat2=" Jan "
40      ;;
41      02)
42      monat2=" Feb "
43      ;;
44      03)
45      monat2=" Mar "
46      ;;
47      04)
48      monat2=" Apr "
49      ;;
50      05)
51      monat2=" May "
52      ;;
53      06)
54      monat2=" Jun "
55      ;;
56      07)
57      monat2=" Jul "
58      ;;
59      08)
60      monat2=" Aug "
61      ;;
62      09)
63      monat2=" Sep "
64      ;;
65      10)
66      monat2=" Oct "
67      ;;
68      11)
69      monat2=" Nov "
70      ;;
71      12)
72      monat2=" Dec "
73      ;;
74    esac
75    echo "$tag$monat2$jahr"
76  }
77
78  del () {
79    if [[ -z $(cat /etc/raddb/users | grep "^$1 ") ]]; then
80      echo "Nutzer $1 existiert nicht."
81      exit;
82    fi
```

```
83    echo "Loesche Nutzer $1, bitte mit yes bestaetigen"
84    read ack
85    if [[ "$ack" == "yes" ]]; then
86      cat /etc/raddb/users | grep "^$1  " >> /etc/raddb/delusers
87      sed "/$1  / d" /etc/raddb/users > /etc/raddb/newusers
88      rm /etc/raddb/users
89      mv /etc/raddb/newusers /etc/raddb/users
90      echo "Nutzer $1 geloescht"
91    else
92      echo "Loeschen abgebrochen"
93    fi
94  }
95
96  ###   Eigentliches Programm       ####
97  if [[ "$#" == 3 && "$1" == "-add" ]]; then
98    add "$2" "$3"
99    exit;
100 elif [[ "$#" == 4 && "$1" == "-add" ]]; then
101   NEWDATE=$(checkdate "$4")
102   if [[ -z $NEWDATE ]]; then
103     echo "Ungueltiges Datum"
104   else
105     add "$2" "$3" "$NEWDATE"
106   fi
107   exit;
108 elif [[ "$#" -gt 0 && "$1" == "-list" ]]; then
109   cat /etc/raddb/users
110   exit;
111 elif [[ "$#" == 2 && "$1" == "-del" ]]; then
112   del "$2"
113   exit;
114 elif [[ "$#" -gt 0 ]]; then
115   echo "Verwendung:"
116   echo "$0 -add user password"
117   echo "$0 -add user password Ablaufdatum im Format 31.12.2014"
118   echo "$0 -del user"
119   echo "$0 -list"
120   exit;
121 fi
122 echo "Starte interaktiven Modus"
123 echo "1 fuer Benutzer hinzufuegen, 2 fuer Benutzer loeschen"
124 echo "3 um aktuelle Benutzer anzuzeigen, alles andere bricht ab
      "
125 read wahl
126 if [[ $wahl == 1 ]]; then
127   ### Hinzufuegen ###
128   echo "Bitte Nutzername eingeben:"
```

```
129   read user
130   echo "Bitte Passwort eingeben:"
131   read pass
132   echo "Soll ein Ablaufdatum angegeben werden? y oder [n]"
133   read expires
134   if [[ $expires == "n" ]]; then
135     add "$user" "$pass"
136   else
137     echo "Datum angeben im Format 31.12.2014"
138     read date
139     newdate=$(checkdate "$date")
140     if [[ -z $newdate ]]; then
141       echo "Ungueltiges Datum"
142     else
143       add "$user" "$pass" "$newdate"
144     fi
145   fi
146 elif [[ $wahl == 3 ]]; then
147   ### Anzeigen ###
148   cat /etc/raddb/users
149 elif [[ $wahl == 2 ]]; then
150   ### Loeschen ###
151   echo "Bitte Nutzername eingeben:"
152   read user
153   del "$user"
154 else
155   echo "Beende Programm..."
156 fi
```