



Learn by doing: less theory, more results

FreeRADIUS

Manage your network resources with FreeRADIUS

Beginner's Guide

Dirk van der Walt

[PACKT] open source*
PUBLISHING community experience distilled

FreeRADIUS

Beginner's Guide

Manage your network resources with FreeRADIUS

Dirk van der Walt

[PACKT] open source 
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

FreeRADIUS

Beginner's Guide

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2011

Production Reference: 1260811

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-849514-08-8

www.packtpub.com

Cover Image by Asher Wishkerman (a.wishkerman@mpic.de)

Credits

Author

Dirk van der Walt

Project Coordinator

Srimoyee Ghoshal

Reviewers

Ante Gulam

Atif Razzaq

Proofreader

Chris Smith

Acquisition Editor

Chaitanya Apte

Indexers

Hemangini Bari

Tejal Daruwale

Development Editors

Kartikey Pandey

Alina Lewis

Graphics

Nilesh Mohite

Technical Editor

Vanjeet D'souza

Production Coordinator

Adline Swetha Jesuthas

Copy Editor

Neha Shetty

Cover Work

Adline Swetha Jesuthas

About the Author

Dirk van der Walt is an open source software specialist from Pretoria, South Africa. He is a firm believer in the potential of open source software. Being a Linux user for almost ten years, it was love at first boot. From then on Dirk spent his available time sharing his knowledge with others equally passionate about the freedom and affordability open source software gives to the community.

In 2003, Dirk started coding with Perl as his language of choice and gave his full attention to functional and aesthetic user interface design. He also compiled an online Gtk2-Perl study guide to promote the advancement of Perl on the desktop.

As Rich Internet Applications (RIA) became more popular, Dirk added the Dojo toolkit and CakePHP to his skills set to create an AJAX-style front-end to a FreeRADIUS MySQL database. His latest work is YFi Hotspot Manager. Today YFi Hotspot Manager is used in many localities around the globe. With many contributors to the project it proves just how well the open source software model can work.

I'd like to thank the Lord Jesus for life and light, my wife Petra and daughter Dani  lle for all their support and understanding, my brother Karel for his interest and help. I would also like to thank the people involved with the FreeRADIUS project, from the coders to the commenters. Lastly I'd like to thank Packt Publishing for supporting Open Source software the way they do.

About the Reviewers

Ante Gulam is a 26-year-old software and system engineer with more than seven years of working experience in various segments of the IT industry. He has worked as a consultant and system engineer on POSIX-compliant systems (Linux, BSD, SCO, and others), and lately has focused mainly on security, design, and administration of Microsoft-based enterprise solutions. Ante is currently working as a system engineer and software developer, primarily on MS platforms (.NET) in Ri-ing d.o.o., a medium-sized software development company.

Being involved in security for several years Ante gained experience in the development of various security tools based on many different technologies and has written articles and co-edited *Phearless Security Ezine* actively for the last four years. Presently, he is working on large networking projects and enterprise environments; adopting them for standards like PCI-DSS enables him to stay in touch with security on the enterprise level.

I would like to thank my family, my friends, and my girlfriend for the their patience. Also all the guys from the "gn00bz" team for all the hours full of fun and knowledge while playing CTF for the past couple of years.

Atif Razzaq holds an MSc degree from Strathclyde University, Glasgow, UK in Communication, Control, and Digital Signal Processing, and a BSc degree in Computer Science from NUCES, Pakistan. After his MSc degree, he started his career as a software engineer in the area of Mobile Application Development in J2ME in Tricastmedia, Glasgow, UK. During this period he also published an article at Java.net titled *Getting Started with BlackBerry J2ME Development*.

He is currently working as the Development Manager at Terminus Technologies who specializes in telecom billing software development. His responsibilities include the development of the billing system and its integration with other applications both proprietary and open source (Asterisk, FreeSwitch, FreeRADIUS, and others). Prior to joining Terminus Technologies, he worked on telecom billing at Comcerto, Bahrain. He has been working on telecom billing and VoIP/SIP Telephony for about three years.

In his free time, he writes his own blog on different ICT topics available at <http://atif-razzaq.blogspot.com>. He can be contacted at atif.razzaq@gmail.com.

It has been a great experience working on this project. I'd like to thank the whole team working on this project: the author and all members from Packt Publishing. I'd like to thank my family for giving up their share of time which I gave to this project. Finally, I'd thank the Great Lord for everything and then my parents who taught me and made me what I am.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read, and search across Packt's entire library of books.

Why Subscribe?

- ◆ Fully searchable across every book published by Packt
- ◆ Copy and paste, print and bookmark content
- ◆ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Introduction to AAA and RADIUS	7
Authentication, Authorization, and Accounting	7
Authentication	8
Authorization	9
Accounting	9
RADIUS	10
RADIUS protocol (RFC2865)	11
The data packet	12
AVPs	15
Vendor-Specific Attributes (VSAs)	16
Proxying and realms	17
RADIUS server	17
RADIUS client	17
RADIUS accounting (RFC2866)	18
Operation	18
Packet format	18
Acct-Status-Type (Type40)	19
Acct-Input-Octets (Type42)	20
Acct-Output-Octets (Type43)	20
Acct-Session-Id (Type44)	21
Acct-Session-Time (Type46)	21
Acct-Terminate-Cause (Type49)	21
Conclusion	21
RADIUS extensions	21
Dynamic Authorization extension (RFC5176)	21
RADIUS support for EAP (RFC3579)	22
FreeRADIUS	23
History	23
Strengths	23

Weaknesses	24
The competition	24
Summary	25
Chapter 2: Installation	27
Before you start	27
Pre-built binary	28
Time for action – installing FreeRADIUS	29
Advantages	29
Extra packages	29
Available packages	30
CentOS	30
SUSE	30
Ubuntu	31
Special considerations	31
Remember the firewall	32
CentOS	32
SUSE	33
Building from source	34
Advantages of building packages	34
CentOS	34
Time for action – building CentOS RPMs	35
Installing rpm-build	36
The source RPM package	36
The package name	36
Updating an existing installation	37
SUSE	37
Time for action – SUSE: from tarball to RPMs	37
Adding an OpenSUSE repository	37
zypper or yast -i	39
Tweaks done by hand	40
Ubuntu	40
Time for action – Ubuntu: from tarball to debs	40
Installing dpkg-dev	42
Using build-dep	42
fakeroot	42
dpkg-buildpackage	42
Installing the debs	43
For those preferring the old school	43
Installed executables	43
Running as root or not	44
Dictionary access for client programs	44
Ensure proper start-up	45
Summary	46

Chapter 3: Getting Started with FreeRADIUS	49
A simple setup	50
Time for action – configuring FreeRADIUS	50
Configuring FreeRADIUS	52
Clients	52
Sections	52
Client identification	53
Shared secret	53
Message-Authenticator	54
Nastype	54
Common errors	54
Users	54
Files module	54
PAP module	55
Users file	55
Radtest	57
Helping yourself	57
Installed documentation	58
Man pages	58
Time for action – discovering available man pages for FreeRADIUS	58
Configuration file comments	60
Online documentation	61
Online help	62
Golden rules	62
Inside radiusd	62
Configuration files	62
Important includes	63
Libraries and dictionaries	63
FreeRADIUS-specific AVPs	64
Running as ...	64
Listen section	64
Log files	65
radiusd	65
Who was logged in and when?	65
Who is logged in right now?	65
Summary	66
Chapter 4: Authentication	67
Authentication protocols	67
PAP	68
CHAP	69
MS-CHAP	70
FreeRADIUS—authorize before authenticate	71

Time for action – authenticating a user with FreeRADIUS	71
Access-Request arrives	72
Authorization	72
Authorize set Auth-Type	73
Authorization in action	73
Authentication	74
Post-Auth	74
Finish	74
Conclusion	74
Storing passwords	75
Hash formats	75
Time for action – hashing our password	76
Crypt-Password	76
MD5-Password	77
SMD5-Password	78
SHA-Password	79
SSHA-Password	80
NT-Password or LM-Password	81
Hash formats and authentication protocols	81
Other authentication methods	82
One-time passwords	82
Certificates	82
Summary	82
Chapter 5: Sources of Usernames and Passwords	85
User stores	85
System users	86
Time for action – incorporating Linux system users in FreeRADIUS	87
Preparing rights	87
SUSE is different	87
CentOS	88
Activating system users	88
Authorize using the unix module	89
Authenticating using pap	89
Tips for including system users	90
MySQL as a user store	90
Time for action – incorporating a MySQL database in FreeRADIUS	91
Installing MySQL	91
Installing FreeRADIUS's MySQL package	92
Preparing the database	93
Configuring FreeRADIUS	94
Connection information	94

Including the SQL configuration	94
Virtual server	94
Testing the MySQL user store	95
Advantages of SQL over flat files	95
Other uses for the SQL database	96
Duplicate users	96
The database schema	96
Groups	97
Using SQL Groups	97
Controlling the use of groups	99
Profiles	100
LDAP as a user store	101
Time for action – connecting FreeRADIUS to LDAP	101
Installing slapd	101
Configuring slapd	102
CentOS	102
SUSE	103
Ubuntu	103
Adding the radiusProfile schema	105
Populating the LDAP directory	106
Installing FreeRADIUS's LDAP package	109
Configuring the ldap module	110
Testing the LDAP user store	110
Binding as a user	111
Advanced use of LDAP	112
Ldap-Group and User-Profile AVP	113
Reading passwords from LDAP	114
Active Directory as a user store	116
Time for action – connecting FreeRADIUS to Active Directory	116
Installing Samba	116
Configuring Samba	117
Joining the domain	118
CentOS	119
SUSE	119
Ubuntu	119
FreeRADIUS and ntlm_auth	119
PAP Authentication	120
MS-CHAP Authentication	121
Summary	122
Linux system users	122
SQL database	123
LDAP directory	123
Active Directory	123

Chapter 6: Accounting	125
Requirements for this chapter	125
Basic accounting	125
Time for action – simulate accounting from an NAS	127
Files for simulation	127
Starting a session	128
Ending a session	129
Orphan sessions	130
Independence of accounting	131
NAS: important AVPs	131
Acct-Status-Type	131
Acct-Session-Id	131
AVPs indicating usage	132
NAS: included AVPs	132
FreeRADIUS: pre-accounting section	133
Realms	133
Setting Acct-Type	133
FreeRADIUS: accounting section	134
Minimising orphan sessions	134
radwho	134
radzap	134
Limiting a user's simultaneous sessions	135
Time for action – limiting a user's simultaneous sessions	135
Session section	137
Problems with orphan sessions	138
checkrad	138
Limiting the usage of a user	138
30 minutes per day in total	139
How FreeRADIUS can help	139
Time for action – limiting a user's usage	140
Activating a daily counter	140
Terminating the session at a specified time	141
rlm_counter	142
Using rlm_sqlcounter	144
Resetting the counter	146
SQL module instance	146
Special variables inside the query	147
Empty account records	147
Counters that reset daily	147
Counting octets	148

Housekeeping of accounting data	148
Web-based tools	149
Summary	149
Chapter 7: Authorization	151
Implementing restrictions	151
Authorization in FreeRADIUS	152
Introduction to unlang	152
Using conditional statements	153
Time for action – using the if statement in unlang	153
Obtaining a return code using the if statement	153
Checking if an attribute exists	156
Using logical expressions to authenticate a user	157
Attributes and variables	158
Attribute lists	158
Time for action – referencing attributes	159
Attributes in the if statement	159
Variables	161
Time for action – SQL statements as variables	162
Time for action – setting default values for variables	163
Time for action – using command substitution	165
Time for action – using regular expressions	166
Practical unlang	167
Limiting data usage	167
Time for action – using unlang to create a data counter	167
Defining custom attributes	167
32-bit limitation	168
Using the perl module	169
reset_time.pl	170
check_usage.pl	172
Installing the perl module on CentOS	173
Updating the dictionary files	174
The recommended way of updating dictionaries	174
Preparing the users file	174
Preparing the SQL database	175
Adding unlang code to the virtual server	175
The SUSE and Ubuntu bug	176
Pre-loading Perl library	177
Testing the data counter	177
Clean-up	178
Summary	179

Chapter 8: Virtual Servers	181
Why use virtual servers?	181
Defining and enabling virtual servers	182
Time for action – creating two virtual servers	183
Available sub-sections	184
Enabling and disabling virtual servers	185
Using enabled virtual servers	185
Time for action – using a virtual server	186
Including a virtual server	186
Handling Post-Auth-Type correctly	187
Taking care of Type attributes	187
Virtual server for happy hour	188
Time for action – incorporating the Hotspot Happy Hour policy	189
Enabling the Happy Hour virtual server	189
Adding the virtual server to a client	190
Defining clients in SQL	191
Consolidating an existing setup using a virtual server	191
Time for action – creating a virtual server for the Computer Science faculty	191
Consolidation implementation	192
A named files section	192
A virtual server for the Computer Science faculty	193
Incorporating the new virtual server	194
What about users stored in SQL?	194
When IP addresses and ports clash	194
Local listen and client sections	195
IPv6	195
Listen section → type directive	195
Pre-defined virtual servers	196
Summary	196
Chapter 9: Modules	199
Installed, available, and missing modules	200
Time for action – discovering available modules	200
Locating installed modules	200
Naming convention	201
Adding alternative paths	202
Available modules	202
Missing modules	202
Including and configuring a module	203
Time for action – incorporating expiration and linelog modules	203
Configuring a module	205

Using modules	206
Sections that can contain modules	207
Using one module with different configurations	207
Order of modules and return codes	210
Time for action – investigating the order of modules	210
Access-Request	211
Return codes	211
Some interesting modules	212
Summary	212
Chapter 10: EAP	215
EAP basics	215
EAP components	216
Authenticator	216
Supplicant	217
Backend authentication server	217
EAP conversation	218
EAPOL-Start	218
EAPOL-Packet	219
Practical EAP	220
Time for action – testing EAP on FreeRADIUS with JRadius	220
Simulator	220
Preparing FreeRADIUS	220
Configuring JRadius Simulator	221
Configuring the eap module	223
The user store	224
EAP on the client	225
EAP in production	225
Public Key Infrastructure in brief	226
Creating a PKI	226
Time for action – creating a RADIUS PKI for you organization	226
Why use a PKI?	227
Adding a CA to the client	227
Configuring the inner-tunnel virtual server	228
Time for action – testing authentication on the inner-tunnel	228
virtual server	228
The difference between inner and outer identities	229
Naming conventions for the outer identity	232
Disabling unused EAP methods	232
Time for action – disabling unused EAP methods	232
Message-Authenticator	233
Summary	234

Chapter 11: Dictionaries	235
Why do we need dictionaries?	235
Parsing requests	236
Generating responses	236
How to include dictionaries	237
Time for action – including new dictionaries	237
How FreeRADIUS includes dictionary files	238
Including your own dictionary files	239
Including dictionary files already installed	239
Adding private attributes	239
Updating an existing dictionary	239
Time for action – updating the MikroTik dictionary	240
Finding the latest supported attributes	241
Location of updated dictionary files	241
Order of inclusions	241
Attribute names	241
Upgrading FreeRADIUS	242
Format of dictionary files	242
Notes inside the comments	242
Vendor definitions	242
Attributes and values	243
Name field	243
Number field	243
Type field	244
Optional vendor field	244
Value definitions	245
Accessing dictionary files	245
Summary	246
Chapter 12: Roaming and Proxying	247
Roaming—an overview	247
Agreement between an ISP and a Telco	248
Agreement between two organizations	248
Realms	250
Time for action – investigating the default realms in FreeRADIUS	250
Suffix module	251
NULL realm	251
Enabling an instance of the realm module	252
Defining the NULL realm	252
Time for action – activating the NULL realm	252
Stripped-User-Name and realm	253
LOCAL realm	254
Actions for a realm	254
Defining a proper realm	254

Time for action – defining the realm	254
Rejecting usernames without a realm	256
Time for action – rejecting requests without a realm	256
DEFAULT realm	257
In closing	258
Proxying	258
Time for action – configuring proxying between two organizations	258
Proxying authentication requests	262
Flow chart of an authentication proxy request	263
EAP and dynamic VLANs	265
Removing and replacing reply attributes	266
Time for action – filtering reply attributes returned by a home server	266
Status of the home servers	267
Time for action – using the preferred way for status checking	268
Proxying accounting requests	269
Time for action – simulating proxied accounting	269
Flow of an accounting proxy request	270
Updating accounting records after a server outage	270
Summary	271
Chapter 13: Troubleshooting	273
Basic principles	274
FreeRADIUS does not start up	274
Who's using my port?	275
Checking the configuration	276
Finding a missing module or library	276
Fixing a broken external component	277
FreeRADIUS refuses to start	277
FreeRADIUS runs despite the display of an error message	278
FreeRADIUS only reports a problem when answering a request	278
Using the startup script	279
FreeRADIUS is slow	279
Time for action – performing baseline speed testing	279
Tuning the performance of FreeRADIUS	280
Main server	280
LDAP Module	281
SQL Module	281
Redundancy and load-balancing	282
Things beyond our control	283
FreeRADIUS dies	283

Client-related problems	284
Testing UDP connectivity to a RADIUS server	284
The control-socket virtual server	285
Time for action – using the control-socket and raddebug for troubleshooting	285
CentOS	286
SUSE	286
Ubuntu	286
Using raddebug	287
Remember the log output	288
Spotting a mismatched shared secret	288
Options for raddebug	289
Raddebug auto termination	289
If there's no output from raddebug	289
Authenticating users	290
Editing the users file	290
Using raddebug	291
When passwords change	291
Password length	291
EAP problems	291
The CA certificate	292
Identify where a problem is located	292
Problems with proxying	292
Online resources	293
Using the mailing list	294
Summary	294
Appendix: Pop Quiz Answers	297
Chapter 1	297
Pop quiz – RADIUS knowledge	297
Chapter 2	298
Pop quiz – installation	298
Chapter 3	298
Pop quiz – clients.conf	298
Chapter 4	298
Pop quiz – authentication	298
Chapter 5	299
Pop quiz – user stores	299
Chapter 6	300
Pop quiz – accounting	300
Chapter 7	300
Pop quiz – authorization	300

Chapter 8	301
Pop quiz – virtual servers	301
Chapter 9	301
Pop quiz – modules	301
Chapter 10	302
Pop quiz – EAP	302
Chapter 11	302
Pop quiz – dictionaries	302
Chapter 12	303
Pop quiz – roaming and proxying	303
Chapter 13	303
Pop quiz – troubleshooting	303
Index	305

Preface

FreeRADIUS Beginner's Guide contains plenty of practical exercises that will help you with everything from basic installation to the more advanced configurations like LDAP and Active Directory integration. This book will help you understand authentication, authorization, and accounting in FreeRADIUS using the most popular Linux distributions of today. Larger deployments with realms and fail-over configuration are also covered along with tips. A quiz at the end of each chapter validates your understanding.

What this book covers

The book can be divided into three sections:

1. Introduction and installation (Chapter 1 to Chapter 3)
2. AAA functions of FreeRADIUS (Chapter 4 to Chapter 7)
3. Advanced topics (Chapter 8 to Chapter 13)

Let's see what each chapter deals with:

Chapter 1, Introduction to AAA and RADIUS, introduces FreeRADIUS and the RADIUS protocol. It highlights some key RADIUS concepts, which help the user avoid common misunderstandings.

Chapter 2, Installation, describes how to build and install FreeRADIUS from source on popular Linux distributions. It also covers installing the FreeRADIUS packages included with popular Linux distributions. Ubuntu, SUSE, and CentOS will be used to ensure a wide coverage.

Chapter 3, Getting Started with FreeRADIUS, gives a brief introduction on the various components of FreeRADIUS. It also discusses the process of handling a basic authentication request.

Chapter 4, Authentication, teaches authentication methods and how they work. Extensible Authentication Protocol (EAP) is covered later in a dedicated chapter.

Chapter 5, Sources of Usernames and Passwords, covers various places where username/password combinations can be stored. It shows which modules are involved and how to configure FreeRADIUS to utilize these stores.

Chapter 6, Accounting, discusses the need for accounting and the options available to record accounting data. It also discusses implementing a policy that includes limiting sessions and/or time and/or data.

Chapter 7, Authorization, discusses various aspects of authorization including the use of unlang.

Chapter 8, Virtual Servers, discusses various aspects of virtual servers and where they can potentially be used.

Chapter 9, Modules, discusses the various modules used by FreeRADIUS and how to configure multiple instances of a certain module.

Chapter 10, EAP, a dedicated chapter on EAP, is a one stop for EAP (802.11x and WiFi).

Chapter 11, Dictionaries, introduces dictionaries, which are used to map the names seen and used by an administrator, to the numbers used by the RADIUS protocol.

Chapter 12, Roaming and Proxying, deals with the RADIUS protocol, which allows the proxying of authorization and accounting requests. This makes roaming possible. This chapter covers various aspects of proxying in FreeRADIUS.

Chapter 13, Troubleshooting, works through many common problems, giving examples of what to look for, and how to fix the issue.

What you need for this book

You need to be familiar with Linux and have a solid understanding of TCP/IP. No previous knowledge of RADIUS or FreeRADIUS is required.

To get the most out of the practical exercises you will need a clean install of Ubuntu, SUSE or CentOS

Who this book is for

If you are an Internet Service Provider (ISPs) or a network manager who needs to track and control network usage, then this is the book for you.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Time for action – heading

1. Action 1
2. Action 2
3. Action 3

Instructions often need some extra explanation so that they make sense, so they are followed with:

What just happened?

This heading explains the working of tasks or instructions that you have just completed.

You will also find some other learning aids in the book, including:

Pop quiz – heading

These are short multiple choice questions intended to help you test your own understanding.

Have a go hero – heading

These set practical challenges and give you ideas for experimenting with what you have learned.

Code words in text are shown as follows: "The `rlm_sqlcounter` module allows defining various counters (time or data based) to keep track of a user's usage."

A block of code is set as follows:

```
if(control:Auth-Type == 'PAP') {  
    update reply {  
        Reply-Message := `/bin/echo We are using %{control:Auth-Type}`  
    }  
}
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:


```
if(control:Auth-Type == 'PAP'){
    update reply {
        Reply-Message := `/bin/echo We are using %`{control:Auth-Type}`
    }
}
```

Any command-line input or output is written as follows:

```
INSERT INTO radcheck (username, attribute, op, value) VALUES ('bob',
'Cleartext-Password', ':=', 'passbob');
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "clicking the **Next** button moves you to the next screen".

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.



Downloading the example code for this book

You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Introduction to AAA and RADIUS

It is my pleasure to present you a beginner's guide to FreeRADIUS. This book will help you to deploy a solid, stable, and scalable RADIUS server in your environment.

This chapter is used as an introduction to RADIUS and FreeRADIUS. We will be covering a fair amount of theory and recommend you pay special attention to it. This will supply you with a good foundation on the workings of the RADIUS protocol and will be of much help in subsequent chapters.

In this chapter we shall:

- ◆ See what AAA is, and why we need it
- ◆ Learn where RADIUS started and why it is so relevant today
- ◆ See why FreeRADIUS really shines as a RADIUS server
- ◆ Understand the relationship between AAA, RADIUS, and FreeRADIUS

Let's get started.

Authentication, Authorization, and Accounting

Users gain access to data networks and network resources through various devices. This happens through a wide range of hardware. Ethernet switches, Wi-Fi access points, and VPN servers all offer network access.

When these devices are used to control access to a network, for example a Wi-Fi access point with WPA2 Enterprise security implemented or an Ethernet switch with 802.1x (EAP) port-based authentication enabled, they are referred to as a **Network Access Server (NAS)**.

All these devices need to exercise some form of control to ensure proper security and usage. This requirement is commonly described as **Authentication, Authorization, and Accounting (AAA)**. AAA is also sometimes referred to as the Triple A Framework. AAA is a high-level architecture model, which can be used for specific implementations.

AAA is specified through various RFCs. **Generic AAA Architecture** is specified in RFC 2903. There are also RFCs that cover different AAA aspects.

Authentication

Authentication is usually the first step taken in order to gain access to a network and the services it offers. This is a process to confirm whether the credentials which Alice provided are valid. The most common way to provide credentials is by a username and password. Other ways such as one-time tokens, certificates, PIN numbers, or even biometric scanning can also be used.

After successful authentication a **session** is initialized. This session lasts until the connection to the network is terminated.



Who is Alice?

Alice and Bob are placeholder names. In fact there is a whole character set, each representing a specific role. We will use the following placeholder names:

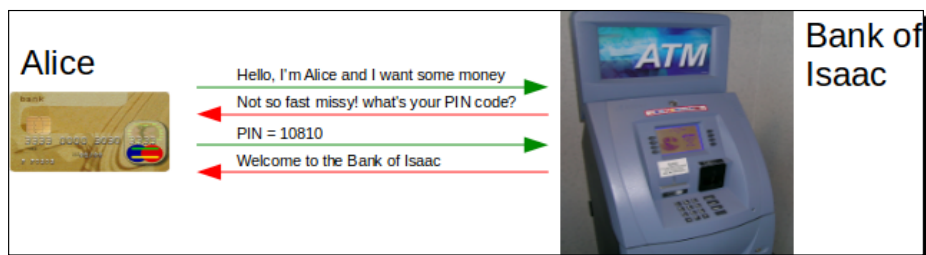
Alice: A user who wants access to our network

Bob: Another user who wants access to our network

Isaac: The Internet Service Provider (ISP)/our network

You can read more about them on Wikipedia: http://en.wikipedia.org/wiki/Alice_and_Bob.

The following image illustrates an authentication process by using the common activity of drawing money from an ATM as an example. This in essence lets you gain access to the bank's network (although it is limited in the extreme).



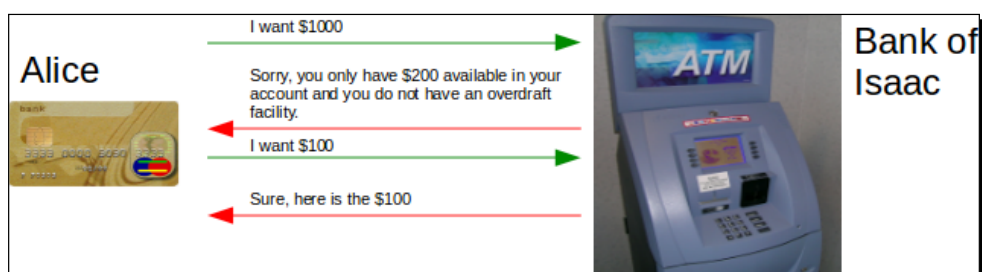
Authorization

Authorization is a means by which Isaac controls the usage of the resources. After Alice has authenticated herself, Isaac can impose certain restrictions or grant certain privileges. Isaac can, for instance, check from which device Alice accesses the network and based on this make a decision. He can limit the number of open sessions that Alice can have, give her a pre-determined IP Address, only allow certain traffic through, or even enforce Quality of Service (QoS) based on an SLA.

Authorization usually involves logic. *If Alice is part of the student group then no Internet access is allowed during working hours. If Bob accessed the network through a captive portal then a bandwidth limit is imposed to prevent him from hogging the Internet connection.*

Logic can be based on numerous things. Authorization decisions for instance can be based on group membership or the NAS through which you connect or even the time of day when you access our resources.

If we take the previous ATM example we can see that if Alice does not have an overdraft facility she will be limited on the amount of money she can withdraw.

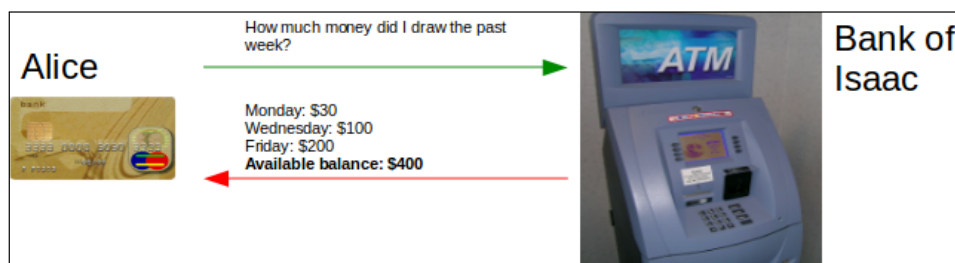


Accounting

Accounting is a means of measuring the usage of resources. After Isaac has established who Alice is and imposed proper control on the established session, he can also measure her usage. Accounting is the ongoing process of measuring usage.

This allows Isaac to track how much time or resources Alice spends during an established session. Obtaining accounting data allows Isaac to bill Alice for the usage of his resources. Accounting data is not only useful to recover costs but it allows for capacity planning, trend analysis, and activity monitoring.

When Alice wants to check her usage and availability of money the ATM offers this functionality. The Bank of Isaac can also monitor her account and discover if she is usually broke before the end of the month. They can then offer her an overdraft facility.



RADIUS is a protocol which is used to provide AAA on TCP/IP networks. The next section will continue with more on the RADIUS protocol.

RADIUS

RADIUS is an acronym for Remote Access Dial In User Service. RADIUS was part of an AAA solution delivered by Livingston Enterprises to Merit Network in 1991. Merit Network is a non-profit Internet provider, which required a creative way to manage dial-in access to various Points-Of-Presence (POPs) across it's network.

The solution supplied by Livingston Enterprises had a central user store used for authentication. This could be used by numerous RAS (dial-in) servers. Authorization and accounting could also be done whereby AAA was satisfied. Another key aspect of the Livingston solution included proxying to allow scaling.

The RADIUS protocol was then subsequently published in 1997 as RFCs, some changes applied, and today we have RFC2865, which covers the RADIUS protocol, and RFC2866, which covers RADIUS accounting. There are also additional RFCs which cover enhancements on certain RADIUS aspects. Having RFCs to work from allows any person or vendor to implement the RADIUS protocol on their equipment or software. This resulted in widespread adoption of the RADIUS protocol to handle AAA on TCP/IP networks. You will find the word RADIUS is used loosely to either mean the RADIUS protocol or the entire RADIUS client/server system. The meaning should be clear from the context in which it is used.

Supporting the RADIUS protocol and standards became the de facto requirement for NAS vendors. RADIUS is used in a wide variety of places, from cellular network providers having millions of users to a small WISP start-up providing the local neighborhood with Internet connectivity to enterprise networks that implement Network Access Control (NAC) using 802.1x to ring fence their network. RADIUS is found in all these places and more!

ISPs and network administrators should be familiar with RADIUS since it is used by various devices that control access to TCP/IP networks. Here are a couple of examples:

- ◆ A firewall with VPN service can use RADIUS.
- ◆ Wi-Fi access points with WPA-2-Enterprise encryption involve RADIUS.
- ◆ When Alice connects through an existing Telco's infrastructure using DSL; the Telco's equipment will use RADIUS to contact Isaac's RADIUS servers in order to determine if she can gain Internet access through DSL (proxying).


The next section will summarize the RADIUS protocol as specified in RFC2865.

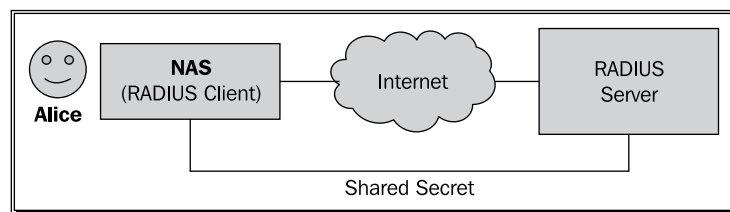
RADIUS protocol (RFC2865)

This section explores the RADIUS protocol on a technical level as published in RFC2865. RADIUS accounting is excluded. This is published as RFC2866 and explored in its own section.

The RADIUS protocol is a client/server protocol, which makes use of UDP to communicate. Using UDP instead of TCP indicates that communication is not strict on state. A typical flow of data between the client and server consists of a single request from the client followed by a single reply from the server. This makes RADIUS a very lightweight protocol and helps with its efficiency across slow network links.

Before successful communication between the client and server can be established, each has to define a shared secret. This is used to authenticate clients.

[ An NAS acts as a RADIUS client. So when you read about a RADIUS client it means an NAS.]



RADIUS packets have a specified format defined in the RFC. Two key components inside a RADIUS packet are:

- ◆ The **code**, which indicates the packet type
- ◆ Attributes, which carry the essential data used by RADIUS

Let's investigate the composition of a RADIUS datagram.

The data packet

Knowing the format of a RADIUS packet will greatly assist in understanding the RADIUS protocol. Let us look more closely at the RADIUS packet. We will look at a simple authentication request. A client sends an Access-Request packet to the server. The server answers with an Access-Accept packet to indicate success.

The RADIUS packets shown here are only the payload of a UDP packet. A discussion of the UDP and IP protocols is beyond the scope of this book.



The screenshots were obtained by capturing the network traffic between the RADIUS client and RADIUS server.

We used a program called Wireshark to capture and look at the content of the data packets. Wireshark is an open source tool that should be part of any serious network guru's arsenal. It can be found here:

<http://www.wireshark.org>

The screenshots here are the result of a simple Authentication request send to a RADIUS server. The obtaining of this data is commonly known as packet sniffing among IT geeks.