WHITEPAPER

Lucent Technologies
Bell Labs Innovations

Lucent Technologies
Remote Access Business Unit

PortAuthority RADIUS White Paper

Table of Contents

1.0 Executive Summary

Remote Access Dial In User Server (RADIUS) was invented by the Remote Access Business Unit of Lucent Technologies in 1992. Since that time access control servers have been designed and customized primarily to manage the technical aspects of remote access. They authenticate user logins, store information pertaining to each user's authorization levels on the network, and collect session information such as time on-line, calling-line identification (or CLID), and modem connect speeds. But, with the rapid growth of the Internet user population in recent years and the accompanying development of new technological capabilities, service providers now need a much more comprehensive remote access management solution—one that will fully integrate technical and business management capabilities.

To meet that need, Lucent Technologies offers a complete remote access management access control solution called PortAuthority™. This family of RADIUS servers provides highly flexible, carrier-class software. PortAuthority includes not only the RADIUS authentication, authorization and session accounting (AAA) capabilities that have become the de facto remote access industry standard but also the sophisticated applications required to manage the business aspects of remote access. The applications support client and departmental accounting, service definition, network analysis for capacity planning, and remote access outsourcing.

2.0 Introduction to RADIUS

In the early days of commercial Internet applications, remote access network administration relied on completely distributed systems. User information was stored on each remote access server (RAS), with little centralized administration capability. Limitations in RAS memory storage space impeded the growth of remote access networks, making it impossible for them to serve the exploding number of users. Moreover, the lack of a central point of control created enormous administrative overhead and prevented effective network security because of inconsistencies in the user information stored in distributed RAS equipment. The first major advance in remote access management was the invention and standardization of the RADIUS protocol. RADIUS introduced a client-server architecture that enabled the efficient centralized management of AAA data.

To satisfy the need for more efficient network scaling, the RADIUS standard designated that all AAA information be stored on central RADIUS servers. All RAS units could then authenticate users and grant them authorization privileges by dynamically accessing a RADIUS server. RADIUS also specified a reliable way to collect session accounting records which could be processed for billing and network analysis purposes.

Based on a model of distributed security earlier defined by the Internet Engineering Task Force (IETF), RADIUS provides an open and scalable client-server security system. The RADIUS server can be easily adapted to work with third-party security products or proprietary security systems. Any network hardware that supports the RADIUS client protocols can communicate with a RADIUS server.

2.1 RADIUS Client-Server Architecture

As a client/server system, RADIUS secures networks and network services against unauthorized access. RADIUS includes three components:

• Authentication and accounting server software running on a centralized computer

• Client software running on remote access servers or other network nodes

• The RADIUS protocol used to communicate between servers and clients

When services are requested of a RADIUS-enabled remote access server, the RADIUS client sends a request to the RADIUS server for authentication of the party requesting the service along with authorization for the specific service(s) requested. In turn, the RADIUS server answers these requests by consulting a data store of user and service policy information.

RADIUS servers may access user authentication and service information maintained in a variety of data formats suitable to the customer's requirements.

Most RADIUS servers support a de facto standard text file format for user information. Alternate sources of user information might be UNIX password files, Sun's Network Information Service (NIS and NIS+), Microsoft Windows NT Domain Directory, X.500 databases accessed via the Lightweight Directory Access Protocol (LDAP), and Structured Query Language (SQL) databases accessed via Open Data Base Calls (ODBC).

2.2 User Authentication with RADIUS

RADIUS authenticates users through a series of communications between the client and the server. Once a user is authenticated and services are authorized, the client provides that user with access to the appropriate network services.

The following steps describe a typical authentication process between a RAS client and a RADIUS Server:

1) Using a modem, the user dials-in to the RAS.

2) Once the modem connection is completed, the RAS either prompts the user for a name and password or, more typically, uses Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP) to request the name and password directly from the user's computer.

3) From this information, the RAS creates a RADIUS data packet called the Authentication Request. (This packet includes information identifying the specific RAS sending the Authentication Request, the port that is being used for the modem connection, and the user name and password.)

4) For protection from eavesdropping hackers, the RAS encrypts the password before Authentication Request is sent to the RADIUS server.

5) The Authentication Request is sent over the network from the RADIUS client RAS to the RADIUS server. [Note: This communication process can be completed over a local area network (LAN) or a wide area network (WAN), thus allowing network managers to locate a RAS remotely from the RADIUS server. If the primary RADIUS server cannot be reached, the RAS can also be configured to route the request to an alternate RADIUS server.]
6) When the Authentication Request is received, the RADIUS server validates the request and then decrypts the password and extracts the user name and other relevant information. (For most users, this information is compared to data retrieved from the appropriate user information system being supported.)

a) The RADIUS server may not have access to the user information directly. In such cases, the request may be forwarded to another RADIUS server that does have the user name and service information. This forwarding process is often called "proxy forwarding" since the RADIUS server acts as a proxy for the RAS.

b) In addition to authenticating the username and password, the RADIUS server performs a policy check against the user's information to ensure that the requested service should be authorized.

c) The RADIUS server may also authorize services by using other data such as the customer's dialed number identification service (DNIS) or the user's "realm" designation (username@realm).

7) If authentication and authorization policy checks are all passed, the RADIUS server sends an Access Accept to the RAS. This packet includes information needed to enable the requested service. (For example, the RADIUS server can send an Access Accept that allows a user to connect to the network with either Transmission Control Protocol/Internet Protocol [(TCP/IP)] or IPX/SPX over the PPP [Point-to-Point Protocol]. At the same time, the Access Accept can prevent that same user from connecting to the network with the SLIP [Serial Line Internet Protocol]. The Access Accept can even contain filtering information that limits the user's access to specific resources on the network.)

8) If policy conditions are not met at any point in this process, the RADIUS server sends an Access Accept to the RAS to deny access to the network.

9) To prevent response from unauthorized agents on the network, the RADIUS server and RAS use a shared secret key to encrypt password information.

 Copyright © Lucent Technologies, Inc.

10) Once it has received the Access Accept message, the RAS can deliver the appropriate network services to the user.

2.3 Benefits of RADIUS Client-Server Architecture

The RADIUS approach to network security and service authorization provides a number of benefits for Lucent Technologies customers, including the following:

1) Greater Security. The RADIUS client-server architecture allows all security information to be located in a central database, instead of scattered around a network in many different devices. This approach increases security because of centralized control and the consistent administration of access policies.

2) More Scalable Networks. For the same reasons that RADIUS increases security, it also helps remote access networks scale in size. The many distributed RASs can authenticate users and authorize services against a single or a few RADIUS servers. This feature is particularly important for networks that include many POPs (Points of Presence – modem pool locations) with large concentrations of RASs. Since RADIUS is an IETF-proposed standard with virtually universal adoption by remote access and other networking vendors, customers can feel confident that all their deployed networking devices will be able to leverage the RADIUS architecture.

3) Lucent RADIUS Standards Leadership. In addition to supporting the RADIUS client protocols in all of its RAS products, Lucent Technologies chairs the RADIUS IETF working group and co-chairs the current AAA IETF working group.

3.0 PortAuthority RADIUS

To further support mission critical and commercial Internet applications, Lucent has developed the PortAuthority family of RADIUS servers. PortAuthority is based on the Lucent Technologies PolicyFlow℠ architecture. As shown in Figure 1, the foundation of the PortAuthority product is a core RADIUS AAA server module that manages the fundamental access management tasks. Extensible, plug-in software modules enable the construction and management of specific policies that integrate into an existing management infrastructure.

3.1 Implemented in Java

Because PortAuthority is written in the Java programming language, it offers excellent cross-platform portability and allows for the rapid development of new features and custom modules. Java also makes multi-threaded applications much easier to write and maintain, thus ensuring that the available processing power is used efficiently.

3.2 Expandable Plug-in Architecture

Most of the functionality of PortAuthority is contained in plug-in modules. Plug-in modules (or "plug-ins") are Java class files. These class files can perform a complete function, call other class-files, or call external object code through the Java Native Interface (JNI).

The Application Programming Interface (API) allows the assignment of properties to each plug-in. The

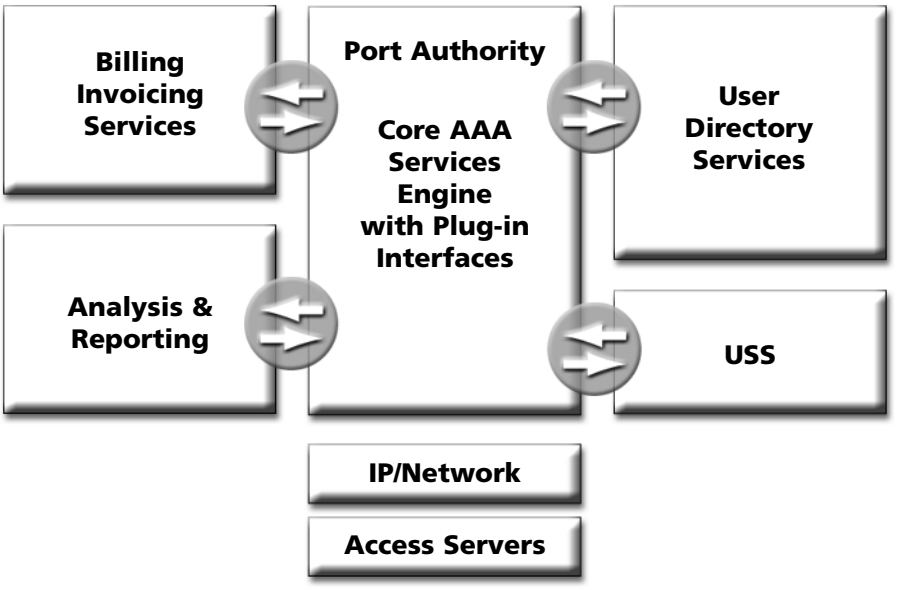

Figure 1. PortAuthority Server and Plug-in Modules

combination of a plug-in and a set of properties is called a "PortAuthority method." Most properties are unique to the plug-in itself. However, a set of standard properties allows control of the logical flow between plug-ins. The "MethodNext" and "MethodOnFail" properties can define logical branching. By combining different sets of properties with a single plug-in (that is, one Java class), it is possible to define multiple PortAuthority methods, all based on the same Java code. Thus, a single plug-in designed to read text files (such as the FilePass files supplied with PortAuthority) can read any number of different file names that have differing formats.

3.3 Flexible PolicyFlow Design

The plug-in design feature of PortAuthority enables the construction of specific logical paths to create a PolicyFlow architecture like the one shown in Figure 2. A few simple building blocks *(PortAuthority methods)* can therefore be chained together to apply complex policies and solve difficult access control problems.

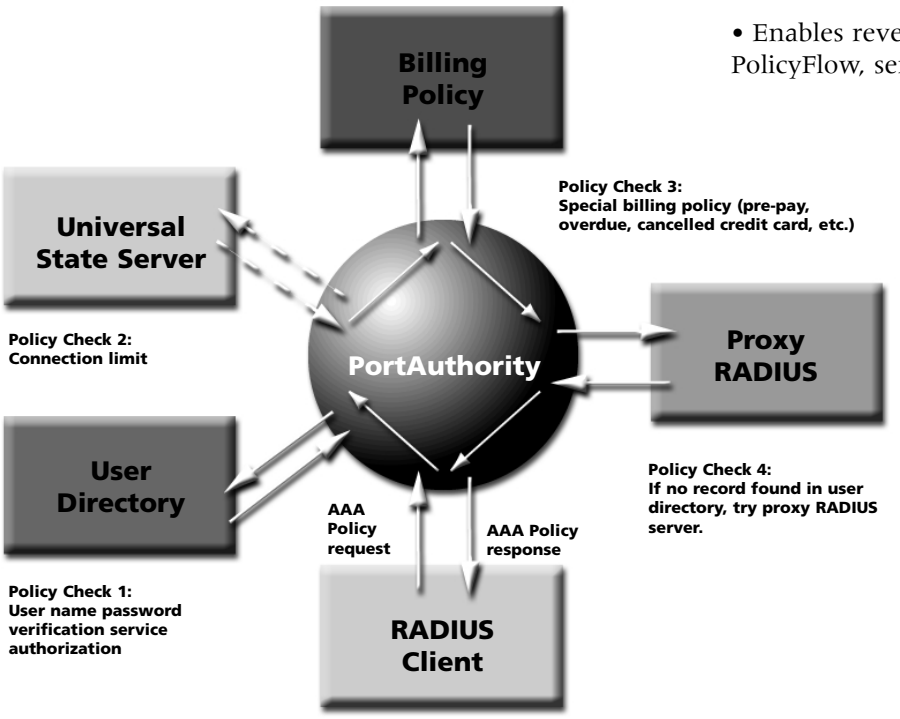

Figure 2. PortAuthority PolicyFlow Architecture

For example, an Internet Services Provider (ISP) may decide to switch from using the UNIX password file as a source of user data to implementing LDAP. Normally, this would require that all users be switched at once in a "slash cutover." With PortAuthority RADIUS, however, a PolicyFlow can be created that looks for a user in the LDAP directory first and, if the user record cannot be found ("MethodOnFail"), the PolicyFlow then tries the UNIX password file. This allows the ISP to do a controlled migration rather than a slash cutover.

Other plug-ins enable special logging options, the re-writing of access request options, the testing user names against a "stop-list," etc. Since proxy (request forwarding) is also a plug-in, it is possible to create a PolicyFlow that looks for a user in the local data source and then, if the user is not found, forwards the request to a roaming consortium for possible authentication.

PolicyFlow can also be applied to the processing of accounting requests. A plug-in to forward accounting records to a remote *(proxy)* location, for example, can call another accounting plug-in to save a local copy using the "Next" property.

The PolicyFlow architecture offers major benefits to service providers, including the following:

• Enables revenue generating new services. With PolicyFlow, service providers can capitalize on new service revenue opportunities. PortAuthority is the ideal to deliver both standard applications such as dial-access and remote-access outsourcing and custom applications and integration that help a service provider compete effectively for revenue and market share.

• Adapts to and helps migrate existing network and systems infrastructure. The flexibility and modularity of PolicyFlow allow PortAuthority to fit easily into the systems that run business today while aiding migrations between legacy and new technologies.

3.4 Universal State Server

To guard against abuses in resource usage, PortAuthority RADIUS server products feature the universal state server (USS). The USS tracks the number of concurrent ports in use for a single user, for user groups identified by a realm designator, or for a Dialed Number Identifica-

 Copyright © Lucent Technologies, Inc.

tion Service (DNIS) provided through remote access outsourcing (RAO).

3.5 Management Interfaces

PortAuthority RADIUS supports the following three management interfaces:

1) Command Line Interface (CLI). The CLI allows the server to be reset and has access to the internal statistical measurements—a sub-set of the RADIUS server Management Information Base (MIB). CLI commands can easily be run from the system scheduler to collect performance data at timed intervals.

2) Telnet Interface. An internal telnet server can listen on a user defined port and allow remote logins (controlled by user name and password). All of the features of the CLI are available through the telnet interface.

3) HyperText Transfer Protocol (HTTP) Interface. PortAuthority includes a built-in password HTTP server that can support standard web pages and a limited common gateway interface (CGI). MIB variables can be referenced in web pages and the actual value of these variables output to the browser. The limited CGI capability allows for the execution of highly specific programs and scripts (shell and perl). These scripts can be used to scan logs, use the PortAuthority test client, perform test authentications, or perform almost any other server-related function.

4.0 PortAuthority Application Samples

The PortAuthority RADIUS server can be used in a wide variety of applications that call for the centralized control of access services and usage accounting. Typical application scenarios include the following:

1) Enterprise telecommuting and world-wide remote access. The enterprise operator provisions a RAS to allow for up to 48 analog or 23 two-channel ISDN users. These users would call the local access number. User accounts would be kept in the corporate human resources (HR) database and a custom PortAuthority plug-in provided to access that HR data. The enterprise would also join a roaming association or consortium like iPass or GRIC. Travelling employees would be provided with a remote dialer program and access lists for the countries where they travel. The local ISP in each country would forward (proxy) the access request to GRIC or iPass. Then, GRIC or iPass would forward the request to the enterprise where the user would be authenticated against the HR database. An Access Accept packet would be sent back to the foreign ISP and the employee allowed online. The ISP would then use a tunneling protocol, like L2TP (Layer Two Tunneling Protocol) to connect back to the enterprise LAN over the public Internet.

2) ISP customized services. A small-town ISP wishes to compete with national ISPs by offering custom services that the bigger competition cannot manage. For example, by assigning each local business with a unique realm (like 'eds-books', 'sues-sandwich-shop', etc.), users at those businesses can be provided with meaningful user names (and e-mail addresses). An address like jack@bay-books is much friendlier than jack12952@bigisp. The ability to manage several data sources and key each one to a realm allows PortAuthority to meet this and similar situations.

3) ISP mergers and acquisitions. A holding company or growth-minded ISP buys several mid-sized ISPs to kick-start its entrance into the Internet access business. Each ISP has its own database of users, in different data stores such as NIS+, Classic RADIUS user files, and LDAP. Disrupting each ISP's user base by requiring a change of name is not acceptable, nor is the situation of completely autonomous RADIUS authentication administration. However, there are numerous overlaps of previously unique usernames. PortAuthority RADIUS servers deployed across the network infrastructure, with the proper PolicyFlows configured, will try all available user data sources serially to find an authentication match for a user login in order to ease integration of overall systems administration.

4) ISP change in data storage technology. An ISP in operation for three years has 140,000 user records stored in and accessed via the Sun NIS+ directory system. For various reasons, the ISP technical management decides to move to LDAP. Current on-line registration, customer care, and billing functions are linked to the NIS+ system; but the LDAP system needs to be tested in production without performing a one-time cutover to ensure survivability of the network services. PortAuthority PolicyFlows can be configured to authenticate users first against the newer LDAP user directory. Only if the PolicyFlow can't find a user entry, does it then attempt to authenticate against the older but more predictable NIS+ database.

5) Billing systems integration. A telco has been operating dial-up Internet services on a completely

different billing system from that of its telephony customers and wishes to offer a single bill for many services, including Internet access. To do this, the telco requires that RADIUS accounting records be sent in real time to its bill rating system. PortAuthority can be adapted with custom plug-ins developed by Lucent or by the telco's development staff to accomplish this integration, without destabilizing the reliability of the PortAuthority RADIUS authentication function.

6) Arbitrage revenue recovery. A telco that provides 83 percent of all outbound calls to the Internet and only 13 percent of the inbound call termination to the Internet is incurring significant costs for Internet access call terminations by Competitive Local Exchange Carriers (CLECs). The telco's management aims to drastically increase its market share of data-dial call termination by rolling out a broad range of new services in a short time frame. The telco's RADIUS development group is overwhelmed by the demand from marketing because their existing RADIUS server solution is inflexible. They need to roll out source code "hacks" to accomplish each new service. But this effort increases instability. PortAuthority allows many services to be rolled out with simple PolicyFlow configurations. It also allows developers to spend more time on truly value-added plug-in customization work.

7) Remote access outsourcing. A major service provider (SP) can purchase inter-office trunks from the local telco for much less than typical Primary Rate Interface (PRI) charges. The SP feels it can install a large number of high-density concentrators (like the PortMaster® 4) and sell "ports" to ISPs, using aspects of scale and low-cost access lines to provide a cost-effective service. Connection between the SP and ISP would be over Layer Two Tunneling Protocol (L2TP) tunnels. The SP would install PortAuthority and assign each contracting ISP a unique telephone number. Calls to an ISP's number would be checked against pre-set pool limits (maximum number of simultaneous users) before the call is answered. If the pre-set limit had been reached, the caller would receive a busy signal triggered by the RAS call-check feature. Otherwise, the call would be answered and the user's PPP session forwarded to the contracting ISP for user authentication against the ISP's own RADIUS server. PortAuthority can also set up tunnels based on the user' realm or calling number identification (CLID). The PortAuthority Universal State Server enforces pool limit policies.

5.0 Sizing PortAuthority RADIUS

RADIUS server loading is usually calculated in AAA transactions per second (called "triple-As"). An AAA transaction consists of one authentication access request, one accounting "Start" record, and one accounting "Stop" record. A PortAuthority RADIUS server, using a text file for user information, can process 40 AAAs per second.

To determine your RADIUS server load, use the following formula.

TPH = "Max occupied busy hour ports"
* (60 minutes / average session length)

AAA Load = TPH / 36000

[3600 is the number of minutes in an hour)

For example: If you operate 25,000 ports with 100% peak hour busy and an
average session length of 20 minutes, you would get these results:

AAA_Load = 25,000 * 2.4 / 3600

AAA_Load = 16.6 AAAs per second

For most installations, we recommend running at least two PortAuthority RADIUS servers and that the AAA load on each server not exceed 1/2 the maximum load the server can handle. By specifying maximum 50% loading, if one server should fail, you ensure that the second server is still capable of handling a full peak busy-hour load. In a two-server RAS configuration, the first PortAuthority RADIUS server with 50 percent of the total load should be designated as 'primary,' and the second with 50 percent of the total load should also be designated as 'primary.' This provides load sharing between the two servers.

6.0 Further Information

For more information on PortAuthority and consultation on designing the optimal RADIUS authentication and policy authorization system for your users and services, contact your local Lucent Technologies sales office, or visit the PortAuthority product web site at: http://www.livingston.com/marketing/products/port-authority.html

 Copyright © Lucent Technologies, Inc.

Lucent Technologies
Remote Access Business Unit
4464 Willow Road
Pleasanton, California 94588-8519
Tel: (925) 730-2500 Fax: (925) 730-2510
website: http://www.lucent.com/dns/portmaster

Copyright © 1999 Lucent Technologies, Inc.
Printed in the USA 06/99 LT-DAT062199

Lucent Technologies
Bell Labs Innovations