Radius Authentication Service

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893. This work is in the public domain, and cannot be copyrighted.

1 Overview

In this lab you will configure a Radius server to handle authentication services for a network device that is already configured to use Radius-based authentication. The Radius server is pre-configured to support an existing network device. You are simply required to add the second device. In this lab, the Radius protocol is configured to use a shared secret known to the Radius server, and the devices that authenticate via that server. The shared secret for both devices is the same.

You are encouraged to use Wireshark within the lab to observe the Radius protocol exchanges.

Radius provides Authentication, Authorization and Accounting (AAA) functions. The protocol is rich and supports many options. Details of the protocol can be found at: `https://tools.ietf.org/html/rfc2865`. While Radius orignally was used to manage AAA for dial-up facilities, it is broadly used for networked devices, including embedded systems such as smart power supplies.

This lab will touch on the basics of using Radius for centralized authentication, and it will demonstrate and example of Radius accounting. The lab uses the FreeRADIUS product, which is detailed here `https://networkradius.com/doc/3.0.10/introduction/RADIUS.html`

1.1 Background

The student is expect to have separately learned about the basic elements of authentication and the Radius protocol.

The student is expected to have at least a basic understanding of the Linux command line, the basics of the file system, and the ability to edit a file. The student should have knowledge of the use of Wireshark, e.g., see the "wireshark-intro" lab.

2 Lab Environment

This lab runs in the Labtainer framework, available at http://nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer radius
```

A link to this lab manual will be displayed.

3 Network Configuration

This lab includes two simulated power distribution control devices that are configured to authenticate users via the Radius protocol. In the jargon of Radius, the control devices are Network Access Server (NAS), i.e., they provide access to the power distirbution functions in our exercise. There are also two client computers

from which users are expected to administer the control devices. And the network includes a Radius server. NOTE: the control devices do not have virtual terminals connected to them, so the only way to access them is through the client computers. The network is illustrated in Figure 1. When the lab starts, you will get three terminals, one connected to each of the client computers and one connected to the Radius server.

The host names of each component are per the diagram. The /etc/hosts files allow use of these host names instead of explicit ip addresses.

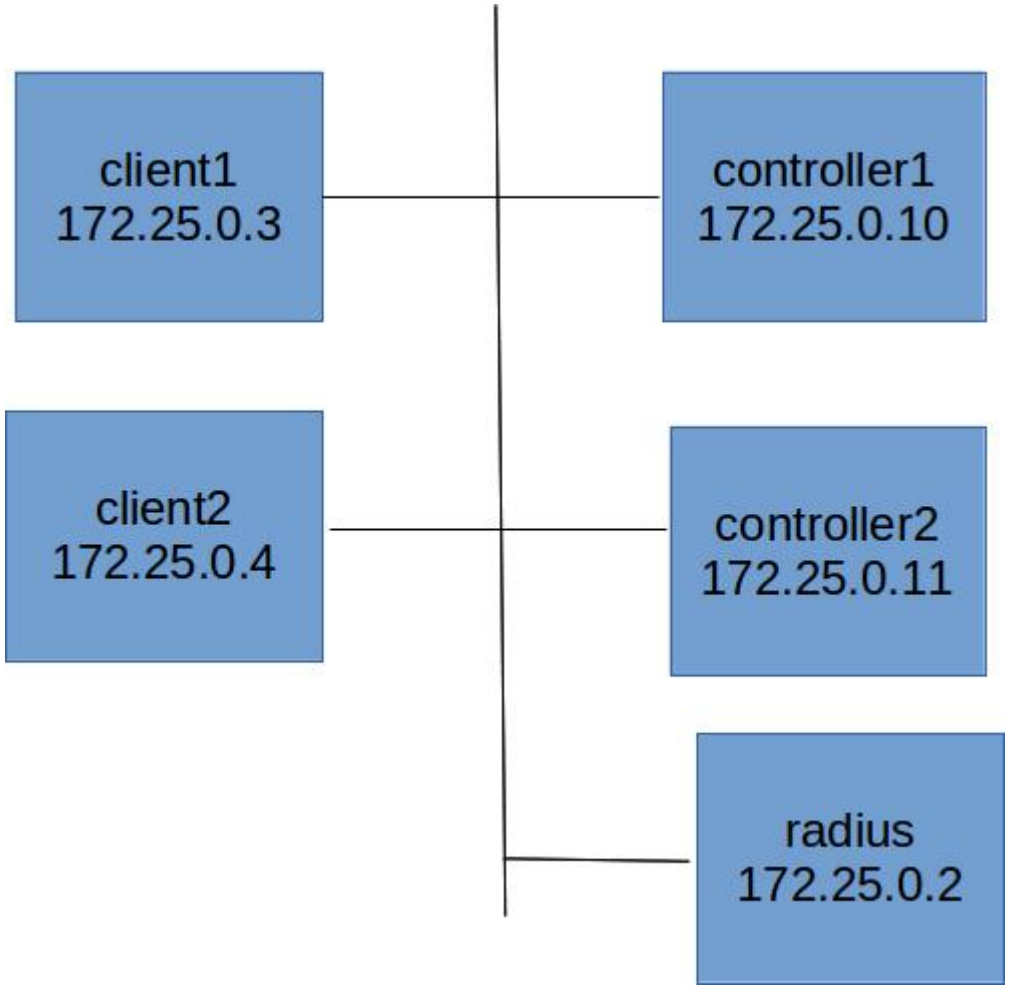

Figure 1: Network topology for the Radius lab

4 Lab Tasks

4.1 Explore

Start wireshark on the radius server:

```
wireshark &
```

and select the eth0 interface and start capturing data.

Then start the radius service in foreground and debug mode:

```
radiusd -fX
```

View the `control_admin` script on client1. Note it simply uses `ssh` to execute a program on a named remote controller. Typically, ssh requires a password or key pairs. The control devices lack password or key management. Instead, control devices are configured to defer ssh daemon authentication decisions to the Radius server.

On client1, connect to controller1:

```
./control_admin controller1
```

When prompted, provide `hardcoded_password` as the password. [1] Observe the traffic in wireshark.

Then use `exit` to exit from controller1. And now try to access controller2, again using a password of: `hardcoded_password`

```
./control_admin controller2
```

What do you observe at the radius service? And in wireshark?

4.2 Configure radius for controller2

The controller2 device has been pre-configured to use your Radius server for authentication of users. That means it has the shared secret used by Radius to encrypt user passwords, and it knows the IP address of the radius server. However, the Radius server is not configured to serve controller2. You must change the Radius server configuration to recognize controller2. Use Ctrl-c at the radius server to stop the service. Edit the `/etc/raddb/clients.conf` file to allow controller2 to authenticate via the radius service, and then restart the radius service.

Try again to access controller2 from one of the clients.

4.3 Change the cadmin password

Stop the radius service and edit the `/etc/raddb/users` file to change the password of the cadmin user to something other than `hardcoded_password`. Then test your abilty use the control_admin utility to access the controllers with the new password.

4.4 Accounting

On the radius server, view the directories and files beneath the `/var/log/radius` directory. Note how each NAS has its own accounting file.

5 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.

[1] If the control_admin program repeatedly informs you that the password is not correct, that may be due to the radius service not running.