Global Information Assurance Certification Paper

Two-Factor Authentication (2FA) using OpenOTP

GIAC (GSEC) Gold Certification

Author: Colin Gordon, colin_gordon@selinc.com
Advisor: Stephen Northcutt
Accepted: September 15th 2014

Template Version September 2014

Abstract

This guide is for security-aware individuals who wish to learn the theory behind user-based two-factor (or multifactor) authentication systems, also known as "2FA". Here we will discuss how 2FA systems work, and how to implement 2FA into a small, virtualized environment for testing purposes. By implementing 2FA, the hope is to enhance the cyber toolkit for administrators who wish to help mitigate the effects of user password theft by cyber intrusion. By following the steps outlined here, the reader should be able to comfortably configure a user account already existing in a Microsoft® Active Directory® (AD) environment to use the Google Authenticator application on his/her smartphone to authenticate with AD username and password+token for remote VPN access.

1. Introduction

A user identity and password alone (known as "single factor" authentication) are not enough to mitigate persistent security risks. Additional methods for proving an identity, besides a secret pin or password, include the use of a biometric (something you are) and/or use of a hardware token or physical key (something you have). The security industry has coined the term *two-factor* (2FA) or *multifactor* for the requirement for two or more different methods of proving an identity (Sophos, 2014). More recently, mainstream digital security systems at large have integrated 2FA methods to reduce the impact of cybercrime and nation-state cyber threats that utilize attacks targeting the discovery, recovery, or theft of user passwords (Davis, 2015). Verizon's 2015 Data Breach Investigation Report (DBIR) recommends 2FA as one of two top mitigation strategies for cyberattacks (Verizon, 2015). Cyber intruders have become so adept at

password theft that, recently, the North American Energy Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) regulations for electrical transmission and generation systems have mandated the use of 2FA for all remote access into any devices that may negatively impact the reliability of the power grid if degraded or misused (NERC, 2014).

The most popular method for enabling the use of 2FA is through the addition of something you *have*, typically in the form of a piece of hardware or a software application on a smartphone, that is carried by the person at all times that generates a random One-Time Passcode (OTP). The user may then use the OTP (a random jumble of numbers and/or letters) in addition to the user's password. To an attacker who has stolen a user's credentials, any pilfered password is now worthless since the authentication system requires the additional OTP information that the user is physically carries at all times.

Many industries and organizations see the risk-reduction benefit of integrating 2FA systems into their user access control environments. However, any system administrator who has dealt with common enterprise 2FA services knows the difficulties that such systems can present. Many enterprise multifactor authentication systems require dedicated maintenance, are expensive; and from a technical difficult perspective, can be difficult to approach for most cybersecurity personnel. However, in addition to paid 2FA services, there are free, easy-to-approach, and easy-to-maintain alternatives (Davis, Two Factor Auth (2FA) Providers, 2015).

1.1. Types of Authentication Tokens

Currently, there are three different OATH OTP types that are the most widely used: event-based tokens, time-based tokens, and challenge-based tokens.

Event-Based Token (HOTP): An OTP system generates event-based tokens on-demand using a combination of a static random key value (HMAC; the H in HOTP) and a dynamic value, such as a counter (IETF, 2005). The event-based token is usually valid for a variable amount of time, but could be valid for an unlimited amount of time.

Time-Based Token (TOTP): An OTP system generates time-based tokens automatically every so often based on a static random key value and a dynamic time value (such as currently time of day). The time-based token is only valid for a certain amount of time, such as 30 or 60 seconds (IETF, TOTP: Time-Based One-Time Password Algorithm, 2011). TOTP is a subset of HOTP.

Challenge-Based Token (OCRA): An OTP system generates challenge-based tokens on demand (IETF, OCRA: OATH Challenge-Response Algorithm, 2011), using a random challenge key that is provided by the authentication server at each unique user log-in. The challenge-based token is valid for a certain amount of time such as several minutes.

1.2. Which 2FA Model Should I Use?

Event-based one-time passcodes (HOTP) may be usable for a long period of time, which increases the likelihood that the OTP could be stolen or misused. Time-based tokens (TOTP) are currently gaining in popularity over event-based tokens (HOTP) due to the additional security TOTP provides via a set, predictable windows of use for one-time passcode (typically 30-90 seconds).

Challenge-based one-time passcodes (OCRA) are a good choice for organizations that want the configuration process for end-users to be as painless as possible, since pushing a challenge token to a user only requires a phone number and/or email address. There are some downsides to the OCRA method, in that it is reliant on cellular or network connectivity between the authentication server and the user to ensure that the authentication process will function. Please see Table 1 for pros/cons of different OTP types.

Table 1 Comparison of OTP Types

	Example	**End-User Configuration Steps**	**Pros/Cons**
HOTP	RSA SecureID® (older versions), Yubikey.	Usually requires hardware token.	Pro: OTP lasts longer for use. Con: OTP is exposed for a longer period, may be valid until next time event-based OTP is generated.

TOTP	Google, Yahoo, LinkedIn, using Google Authenticator.	Installation and configuration of smartphone application.	Pro: OTP is exposed only a short amount of time. Con: User only has short amount of time to type in valid OTP
OCRA	Banking websites that send tokens via SMS text or email.	Little-none; requires submission of phone number or email address.	Pro: Easy end-user configuration. Con: Typically relies on external communication medium (cellular network, internet connection) to get OTP to user

Another decision for organizations implementing 2FA using an OTP system is what type of token client to use (either software-based or hardware). Highly secure environments may demand the use of small hardware devices that generate and display one-time tokens, since the security industry generally recognizes that it is more difficult to extract keys from or compromise hardware tokens. These are so-called "true" 2FA solutions (Sophos, 2014) that do not rely on any mobile network or software platform to get OTP tokens.

However, soft-tokens, OTPs that are generated and displayed by smartphone applications, are generally easier to manage and configure, and may be used with most every smartphone variety, which is a low-cost and end-user friendly decision, especially in organizations that are adopting Bring Your Own Device (BYOD) policies (see Figure 1). Here, we use the Google Authenticator smartphone application as a soft-token manager. Google Authenticator generates time-based one-time passcodes (TOTP).

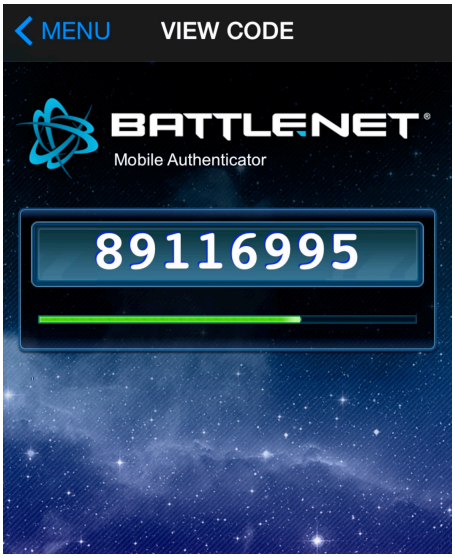

Figure 1 *Battlenet Mobile Authenticator* Smartphone Application TOTP Generator

1.3. Introducing OpenOTP

OpenOTP™ Authentication Server by RCDevs is a highly configurable authentication server that utilizes open-source solutions and systems. OpenOTP is flexible enough to act both as a stand-alone option, using a free user database, or may be integrated into existing Microsoft® Active Directory® or other centralized user directory services or databases. OpenOTP stands out as an approachable method for introducing 2FA by easily enabling the addition of "*something you have*" to existing groups of existing users with passwords.

OpenOTP supports software tokens and hardware tokens, such as Yubikey, FIDO, SecuTech, and others, which can live on a person's smartphone. Additionally, OpenOTP brings the following non-exhaustive list benefits to its administrators and end-users:

- Ability to install onto an existing Linux-based OS on commodity hardware
- Ability to quickly commission and test via pre-packaged and pre-configured Virtual Machines (VMs)
- Comprehensive documentation and support that is freely available on the OpenOTP website
- Ubiquitous smartphone and hardware token support
- Support for hardware security modules (HSMs) for safer storing of secret token information
- Supports Initiative for Open Authentication (OATH) algorithms for better OTP hardware and soft-token interoperability
- Active support and maintenance via RCDevs company developers
- Free licenses for up to 40 active users
- Support for three different OATH OTP types: event-based tokens, time-based tokens, and challenge-based tokens.

For more information about how OpenOTP works, please see Appendix A.

Author Name, email@address

2. Initial Configuration

2.1. Before You Begin

Users completing this guide will require basic to medium Ethernet networking skills. Some knowledge of virtual environments, security concepts, and Linux-based operating systems are helpful but not required.

Here, the author uses a virtualized environment using the following Operating Systems:

- Windows Server (2008 R2)
- Linux with OpenOTP installed. The author used the RCdevs OpenOTP virtual instance with WebADM Web-Based Directory Administrator (download at https://www.rcdevs.com/downloads/index.php?id=VMWare+Appliances#)
- Authentication clients that supports RADIUS. Here the author uses an OpenVPN Access Server virtual appliance (download at https://openvpn.net/index.php/access-server/download-openvpn-as-vm.html)
- Laptop or workstation for testing, configuration, and hosting virtual machines

The author recommends using virtualization for testing environment (e.g. VMware, VirtualBox, HyperV).

2.2. Lab Configuration

The author's lab configuration consists of a Windows Server 2008 R2 as the primary user directory, the OpenOTP server (version 1.3.3-2) for 2FA and RADIUS communications, and then various "test" machines and devices (one Linux and one Windows).

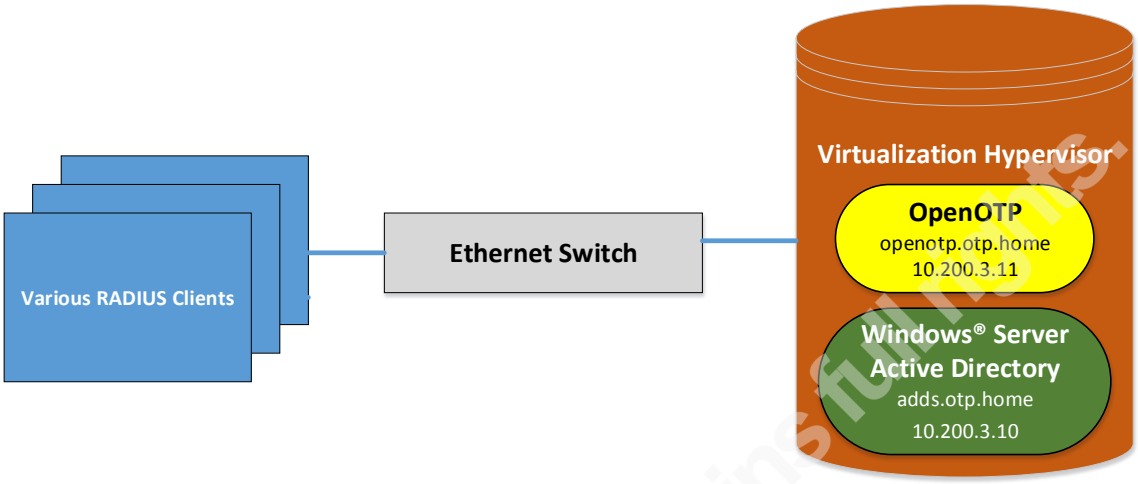

Figure 2 Lab Architecture

The author will test the solution using the OpenVPN Access Server (version 2.0.12 at the time of writing) as an authentication clients.

2.3. Active Directory® (AD) Instance Configuration

Here, the author created a running Windows 2008 R2 instance with Active Directory Domain Services installed and functional. OpenOTP requires the following configurations on the AD server:

- X.509 certificate installed on the Windows Server or AD Domain Services service to protect the confidentiality of user data during LDAP transactions between OpenOTP and the user directory. Please ensure that a certificate is active, and that the AD instance can be accessed securely via the "STARTTLS" method on TCP port 389. NOTE: STARTTLS is not strictly required. However, be aware that unprotected LDAP binds will carry Domain Admin credentials in clear-text over the network. Also, Active Directory requires STARTTLS if you intend to allow OpenOTP to update confidential user data, such as user passwords.
- Several distinct AD groups, with active members. The example of some active users and groups can be see (see Table 2 for users/groups the author will use). Please ensure that the users in the table are members of their respective groups.
- DNS services should be installed and active on the AD server.

Three groups (Engineers, IT_Technicians, IT_Supervisors) and three respective users, “Ada Engineer”, “Ted Technician”, and “Bob Supervisor” were used. Then, OpenOTP Authenticator Server will be used to map the three groups to distinct authorizations on each authentication client (in the cases where authorizations are supported by the authentication clients).

Table 2 Active Directory Group/Privilege Matrix

Group	User	Remote VPN	Network Devices Restricted	Network Devices All
Engineers	"Ada Engineer"	x		
IT_Technicians	"Ted Technician"	x	x	
IT_Supervisors	"Bob Supervisor"	x		x

In addition to the configurations mentioned above, OpenOTP requires access to the Active Directory® server using a user account with Domain Admin level privileges. OpenOTP uses the Domain Admin user account to extend the schema of the AD instance to support the OpenOTP attributes for users/groups and as a proxy user to authenticate to the AD server using LDAP protocol to checks the authenticity of users, accounts, and attribute data.

OpenOTP requires a “Super Administrators” user or group in the Active Directory server. Any member of this group has full access to the WebADM configuration interface. The author will use the “IT_Supervisors” group as the “Super Administrators” for OpenOTP. Alternatively, a user may use the Domain Administrators group (cn=Domain Admins,cn=Users,dc=otp,dc=home) as the Super Administrators group. The author uses the settings and attributes listed in Table 3.

Table 3 Author’s AD Settings and Respective Attributes

Setting	**Attribute (From Author's Configuration)**
Domain Admin User	Administrator
Domain Admin User DN	cn=Administrator,cn=Users,dc=otp,dc=home
Domain Admin User Password	Asdf123$
Domain Admins Group	cn=Domain

Author Name, email@address

DN	Admins,cn=Users,dc=otp,dc=home
Windows Server Fully Qualified Domain Name (FQDN)	adds.otp.home
Super Administrators Group DN	cn=IT_Supervisors,dc=otp,dc=home

NOTE: Readers may (rightly) point out that configuring the OpenOTP LDAP proxy user for Domain Admin authorizations is a potential security weakness. The OpenOTP LDAP proxy user can be a different user on the LDAP directory with more specific permissions. Please see Chapter 21 of the WebADM Manual *"LDAP Permissions"* for more information about how to restrict the authorizations of the OpenOTP LDAP proxy user (and to avoid using the AD Domain Administrator account).

3. Configuring OpenOTP

3.1. OpenOTP Virtual Image Initial Configuration for VMware ESXi

Please see Appendix B for detailed initial configuration steps for the RCDevs virtual appliance, including configuring the server hostname, IP address, changing the root password, and ensuring the virtual appliance can receive accurate time via Network Time Protocol requires (NTP; critical for OATH TOTP accuracy).

1. Download and install the OpenOTP Virtual Appliance (OpenLDAP – OVF) from https://www.rcdevs.com/downloads/index.php?id=VMWare+Appliances#
2. During the initial boot, the Linux system will prompt you to input the server Fully Qualified Domain Name (FQDN), to enter an organizational name, to enable WebADM to be started automatically, to register the WebADM logrotate script, and to generate a WebADM secret key (the key encrypts WebADM information in the user directory service). The author used **openotp.otp.home** as the FQDN, **Self** as the organizational name, and "yes" to all else.

3. From a web browser, navigate to the IP address of the RCDevs appliance to log in to the WebADM configuration platform using HTTPS on port 10000 (see Figure 45) to finish the initial configuration (https://10.200.3.102:10000 in the case of the author). You may use the default root credentials to log in (**root** / **password**).
4. From the WebADM dashboard, you may set the root password, the network IP address, hostname and DNS server(s), and the time service on the server.

3.2. Integrating OpenOTP into Microsoft Active Directory®

In this section, the RCDev appliance WebADM service will be configured with the attributes necessary (see Table 3) for OpenOTP to integrate into Active Directory®.

1. Navigate to /opt/webadm/conf, and place the Certificate Authority (CA) certificate used to sign the adds.otp.home certificate into this directory. This function can be performed by using using Secure Copy (SCP, see Figure 3).

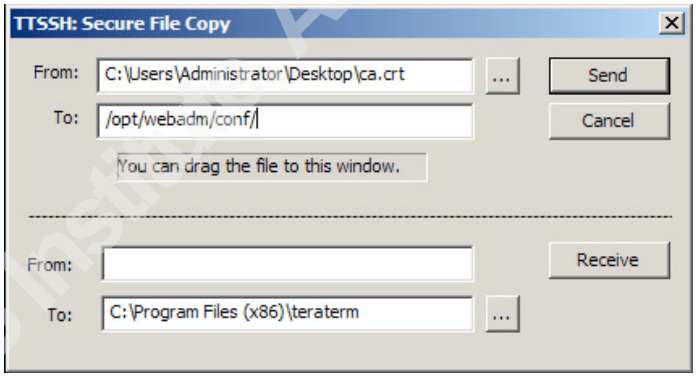

Figure 3 SCP CA Certificate

2. From the /opt/webadm/conf directory, edit **servers.xml** (command: "**vi servers.xml**") and add the hostname for the AD server (**adds.otp.home**), the encryption type ("**TLS**"), and the location of the CA certificate that was just uploaded ("**/opt/webadm/conf/ca.crt**"). Note that the user may need to use the server IP address instead of the hostname if the RCDevs appliance DNS server has not been fully configured (see Figure 4).

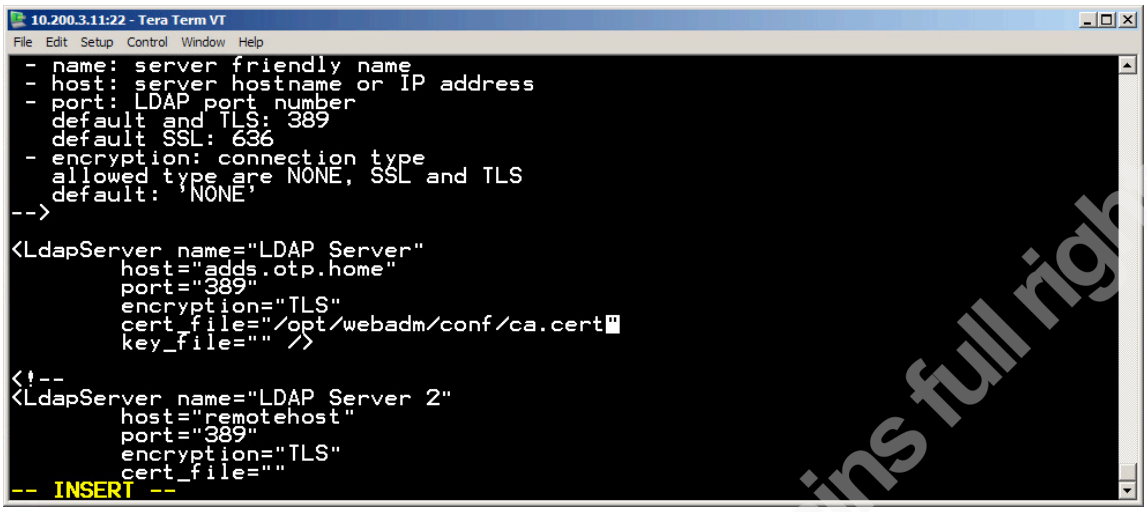

Figure 4 severs.xml

3. Next, edit **/opt/webadm/conf/webadm.conf** to add the AD-specific settings and attributes necessary for OpenOTP to integrate into the Windows® user directory server. Change the **proxy_user**, **proxy_password**, and **super_admins** to match the information in Table 3 (see Figure 5).

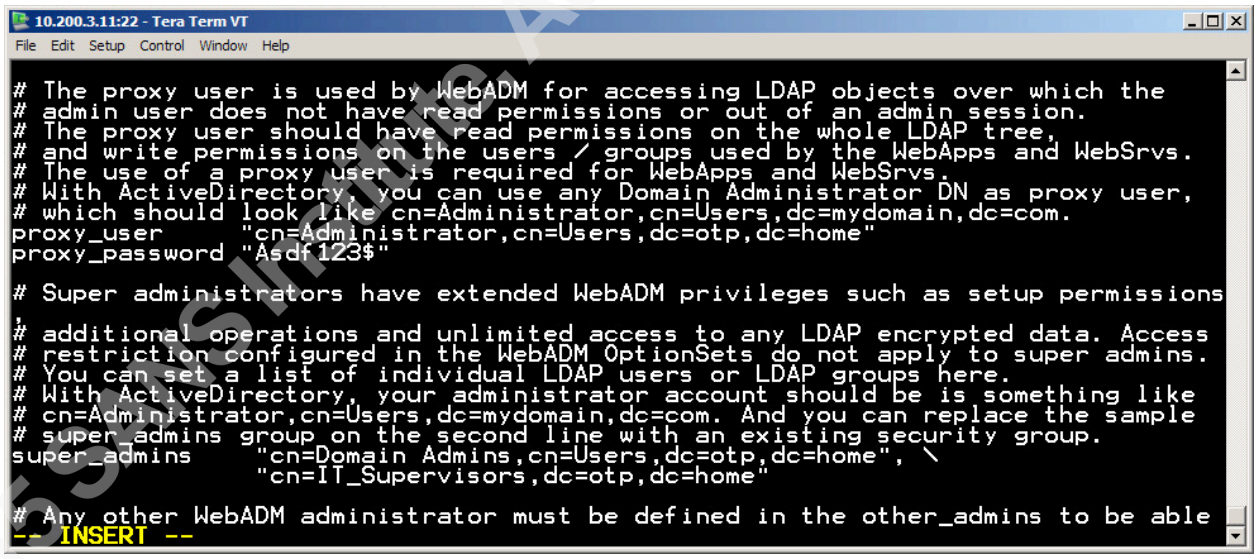

Figure 5 webadm.conf Proxy User and Super Admins Settings

4. Comment out the **other_admins** line (insert a '#' in front of the line).

5. Comment-out the existing LDAP containers required by WebADM, and uncomment-out the Active Directory specific containers. The user will need to edit the "**dc=mydomain,dc=com**" on the end of each line to match the root of your own domain (in the author's case, "**dc=otp,dc=home**"). See Figure 6 for an example.

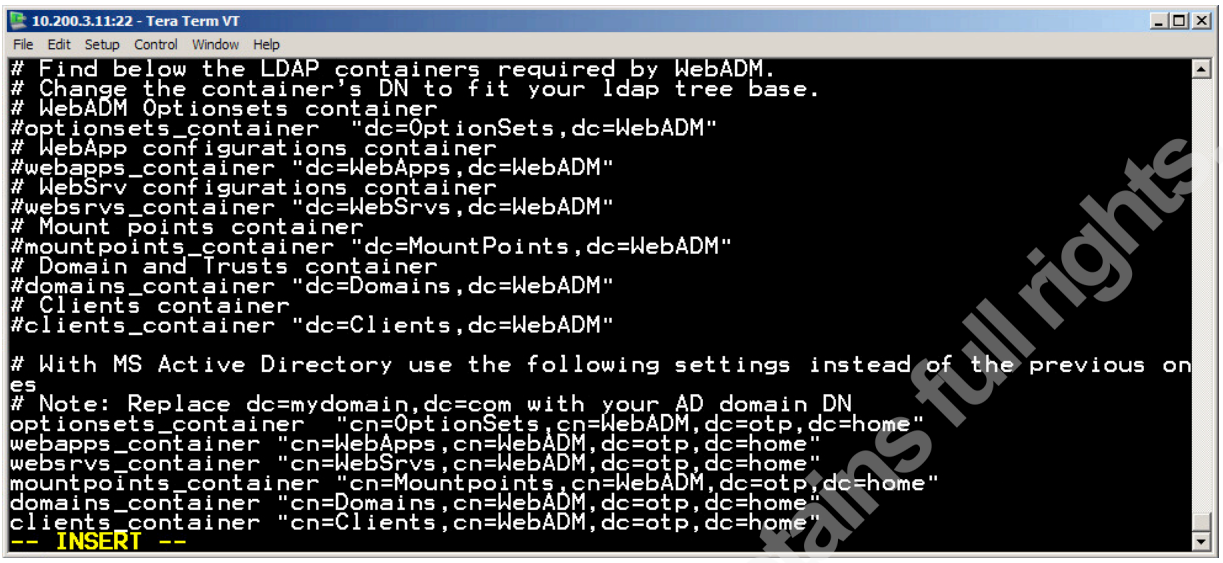

Figure 6 webadm.conf LDAP Container Settings

6. Next, change the **time_zone** setting to match the local time zone. Leave all other settings default. After saving the edited webadm.conf file, the user may restart the webadm service via the **service webadm restart** command. Make sure the **Connected LDAP server** message does not have an error (see Figure 7).

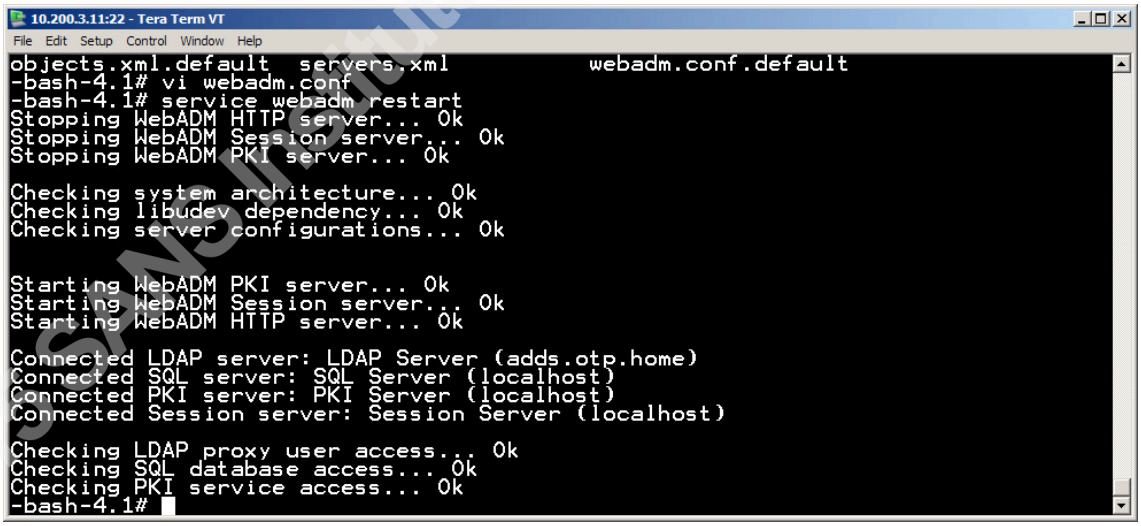

Figure 7 Restarting the WebADM Service

7. Using a web browser, navigate to the IP address of the RCDevs appliance to log in to the WebADM OpenOTP configuration platform using HTTPS to finish the integration of OpenOTP into the Windows® server. Note that during this commissioning phase, the user will need to log in via "UID" mode

(see Figure 8), where the user enters the full DN of your AD Administrative user (in the case of the author, **cn=Administrator,cn=Users,dc=otp,dc=home**).

Figure 8 WebADM UID Login Process

8. After logging in as a super admin, WebADM will prompt the user to run the Setup Wizard (see Figure 9). The Setup Wizard will guide the user to fully integrate OpenOTP into AD, which will both extend the schema for the AD server and push in new WebADM-specific attributes.

NOTE: There is a mode for integrating OpenOTP into the AD server that will work without extending the LDAP schema. To see more about this process, see the WebADM Installation Manual Section 5.4, “Setup the LDAP Directory”. Also note that the Domain Controller the user connects to using OpenOTP does need to be the Schema Master for the Domain.

Author Name, email@address

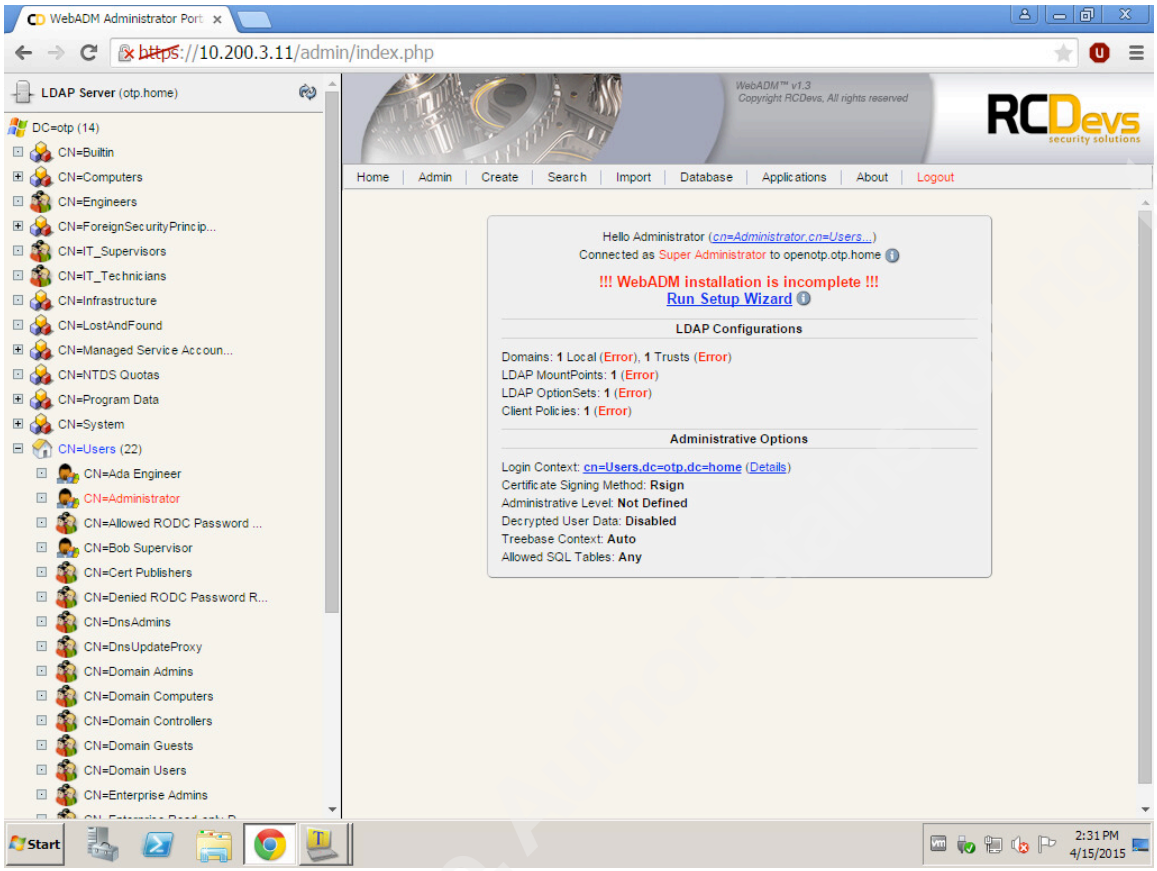

Figure 9 WebADM Setup Wizard

9. Click **Run Setup Wizard**. On the next screen, select **Setup LDAP schema** and **Create default containers and objects**. For both of these functions, WebADM should respond with "Ok" messages (see Figure 10).

Author Name, email@address

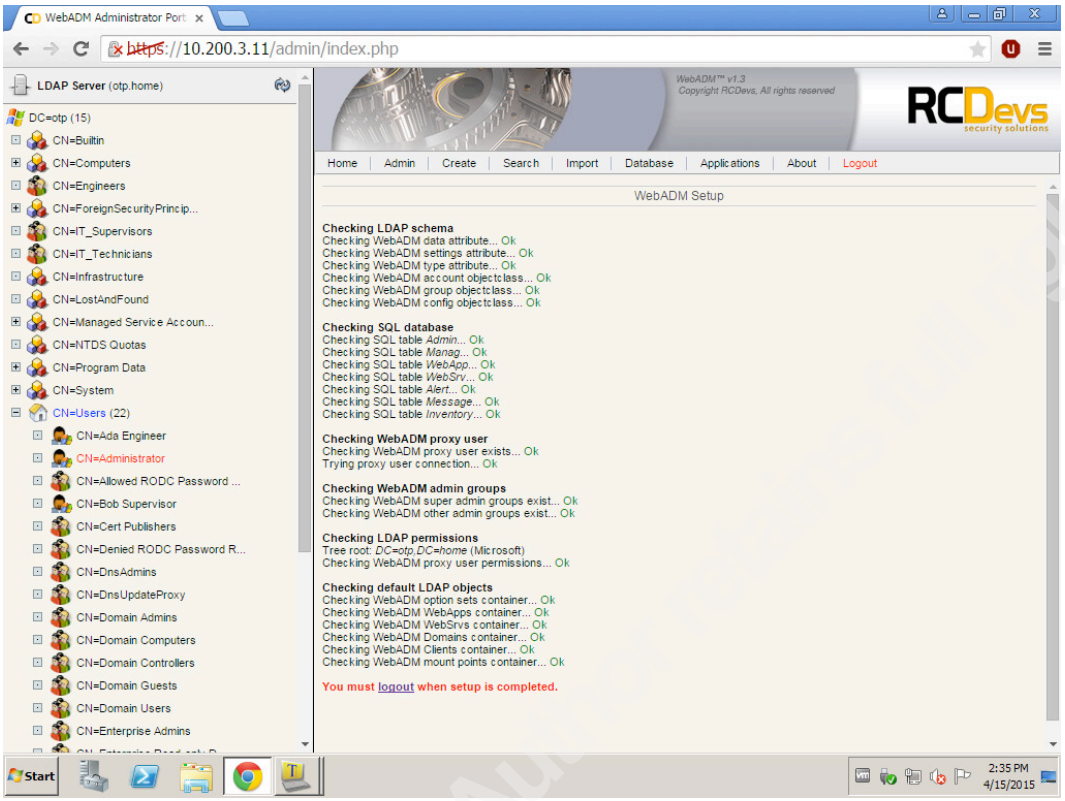

Figure 10 WebADM Setup Wizard Successful

10. After this process is complete, click **Logout**. The user should now be able to log into the WebADM instance using "**Administrator**" (the Domain Admins account) without needing to enter the full user DN (see Figure 11).

Figure 11 WebADM Normal Login Mode

11. Finally, on the WebADM dashboard under **Admin** tab, select **Local Domains**, and click the "**Default**" link. Change the **Object Name** to the name of the domain (**otp.home** in the author's case) and select **Rename** (see Figure 12).

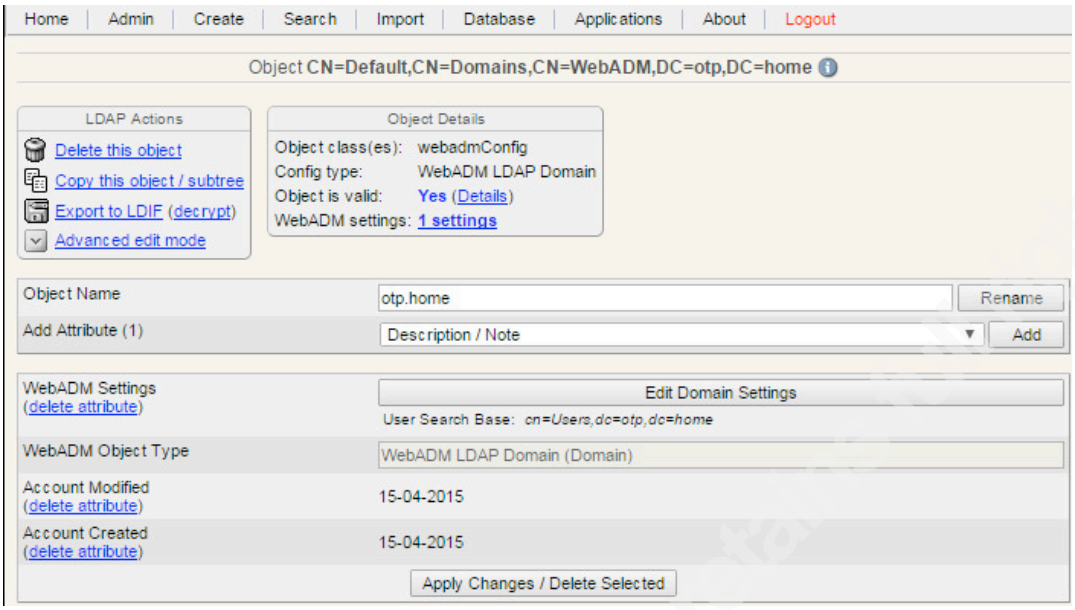

Figure 12 Rename WebADM Domain Object Name

Your OpenOTP instance should now be integrated into the AD server.

3.2.1. Troubleshooting

If there are problems logging into WebADM after configuring /opt/webadm/conf/webadm.conf, the following are some tips:

- Turn on Syslog logging in /opt/webadm/conf/webadm and use the command "**tail –f /var/log/messages**" to see messages sent by the WebADM service in real time.
- Log into WebADM in UID mode using the full DN of a member of the groups listed in the **super_admins** section of webadm.conf. For the Domain Admins, this will be **cn=Administrator,cn=Users,dc=otp,dc=home**, or for the IT_Supervisors this will be **cn=Bob Supervisor,cn=Users,dc=otp,dc=home**.

3.3. Adding Users and Groups to the OpenOTP Authentication Server

Here user accounts and centralized user groups (found in Table 2) will be activated so that the OpenOTP Authentication Server can recognize them.

1. From the WebADM OpenOTP dashboard, select the **Admin** tab, and then go to **Local Domains → Configure**. Click the checkbox (see Figure 13) by

Group Search Base and type in domain root (in the author's case, **dc=otp,dc=home**).

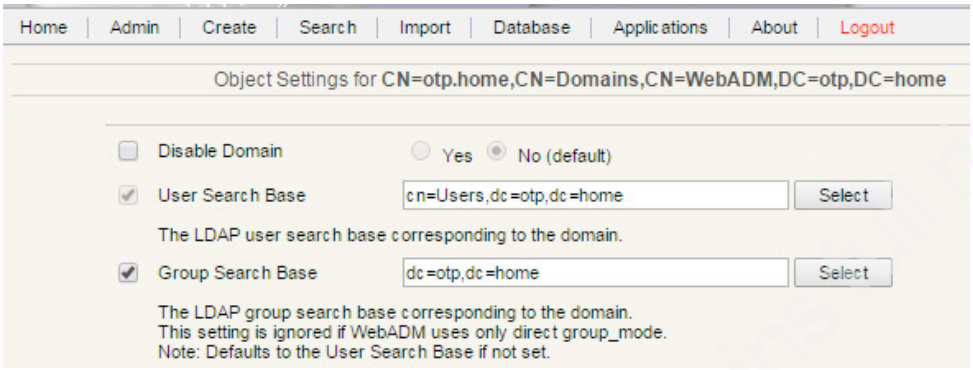

Figure 13 Editing WebADM Group Search Base

2. Next, click the checkbox by **Allowed Groups**. Here, the groups of the wanted members who are going to be able to authenticate using multifactor credentials (as seen in Table 2) need to be selected. To do this, click the **Select** option by the **Allowed Groups** box and select the necessary user groups from the directory navigation panel on the left side of interface (see Figure 14). When finished, click **Apply**.

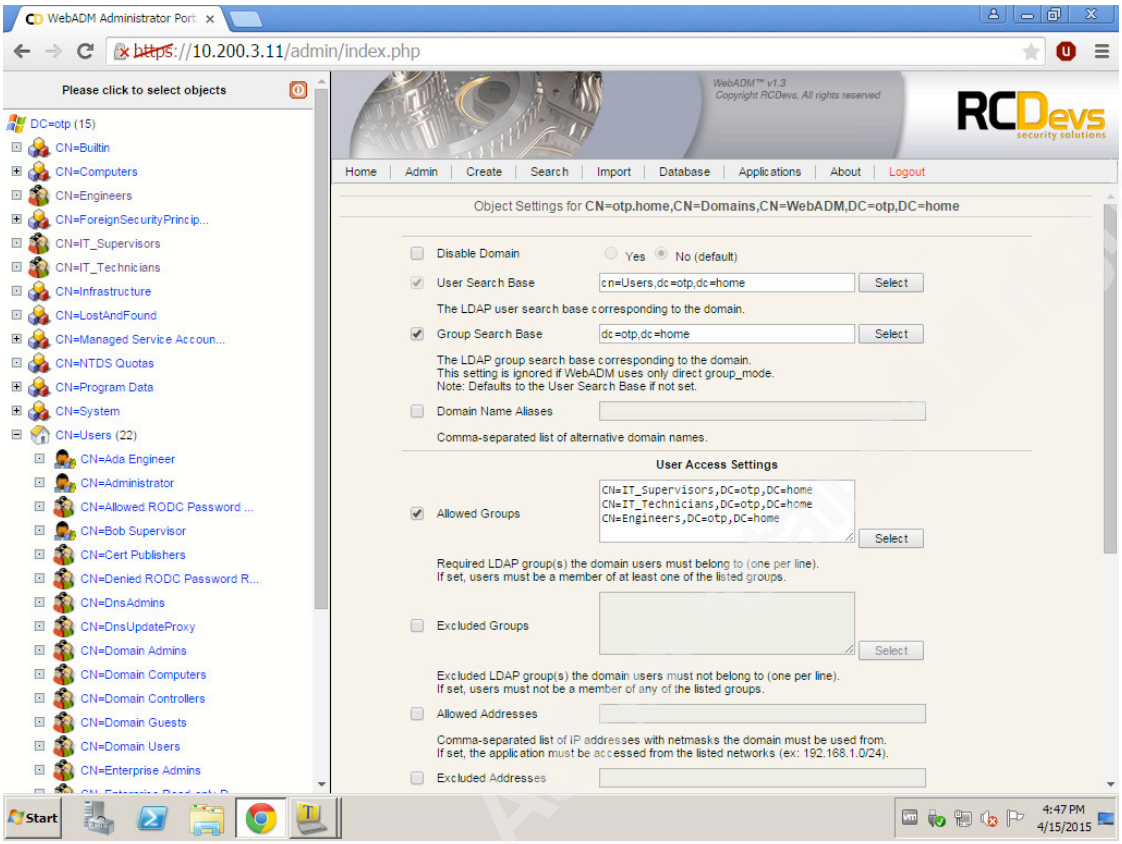

Figure 14 Selecting Allowed Groups in the WedADM Interface

3. Next, the users whom we wish to be able to utilize 2FA must be activated. To do this, select each user (Bob Supervisor, Ada Engineer, and Ted Technician) from the directory navigation panel on the left side of the interface and click **Activate Now** (see Figure 15).

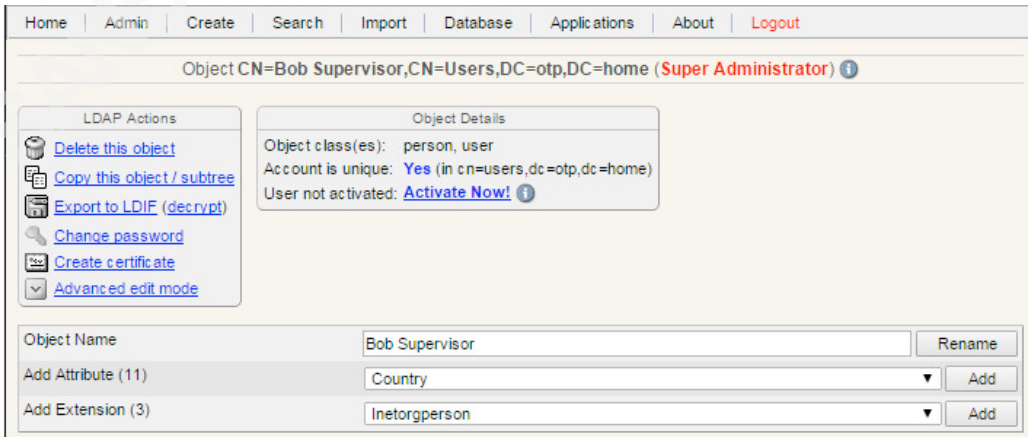

Figure 15 Activating a User in WebADM

4. WebADM will prompt addition of some additional attributes to the user. Click **Proceed** then **Extend Object** to add the default WebADM attributes to the

user account in Active Directory. Note that each user activated within WebADM will count against the bucket of 40 free accounts allowed to use the OpenOTP Authentication Server (see Figure 16).

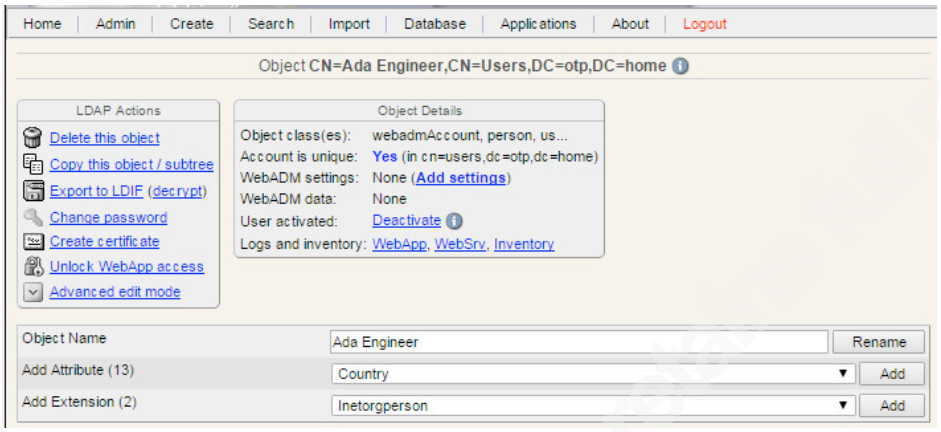

Figure 16 Activated OpenOTP Authentication Server User

By now, various users and groups have been activated to be able to take advantage of the OpenOTP authentication server.

3.4. Testing 2FA with the OpenOTP Authentication Server

In this section, the OpenOTP is enabled to service itself and create TOTPs for the test users. NOTE: This next section requires the **Google Authentication** smartphone application.

1. From the WebADM dashboard, select the **Admin** tab, then **WebApps & WebSrvs**, From the list of **Web Services**, click **REGISTER** under the **OTP & U2F Authentication Server** option (see Figure 17).

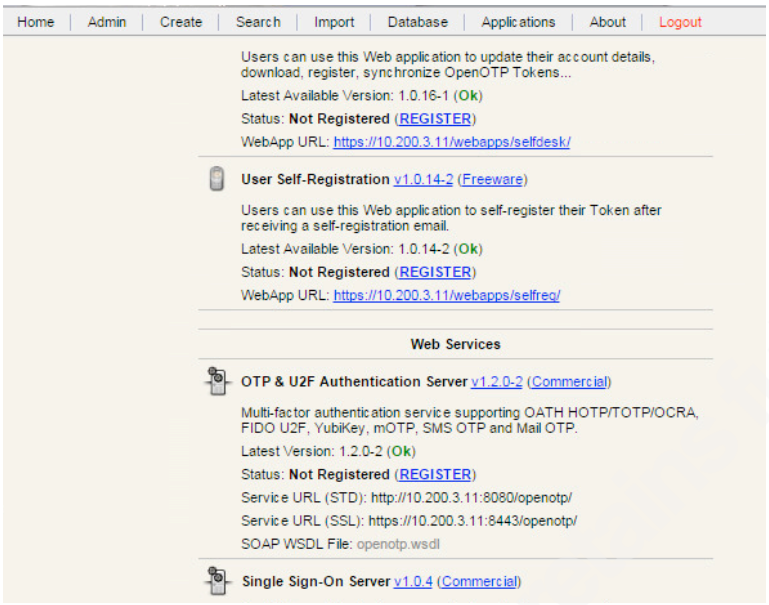

Figure 17 Register the OpenOTP Server

2. After the **OTP & U2F Authentication Server** successfully registers, select **CONFIGURE** on the same service. The Web Service Settings page for the OTP authentication server will provide you with a myriad of options. For now, only change the one (optional) parameter, the **Service Name** (see Figure 18), will be changed. Click the box beside this option and enter the domain name (in the author's case **otp.home**). When finished, click **Apply**.

Figure 18 OTP Service Name

3. Next, add a TOTP configuration for each of our test users. To begin, select the desired user to whom a token from the directory navigation panel should be added. Under the selected user's **Application Actions** should be the **OTP & U2F Authentication Server** option (see Figure 19).

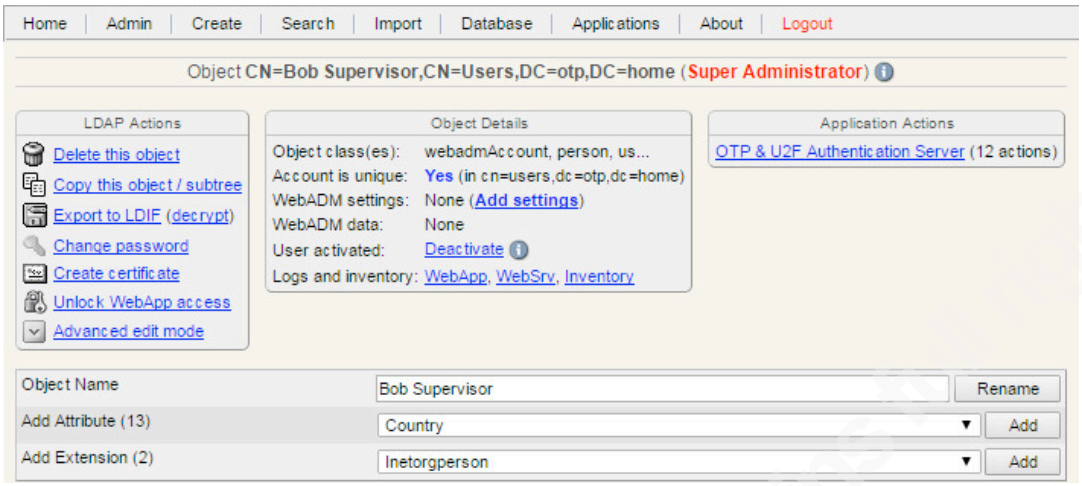

Figure 19 WebADM User Application Action

4. Click the **OTP & U2F Authentication Server** option. On the proceeding screen, there will be a number of options available for the user account (see Figure 20). Select **Register / Unregister OTP Tokens**.

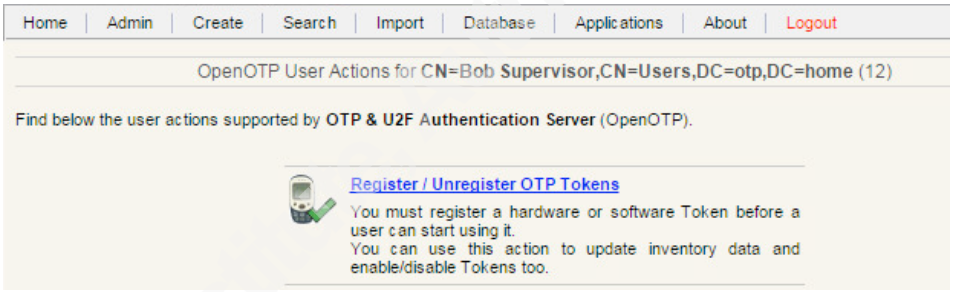

Figure 20 Register WebADM User Token

5. On the Register / Unregister OTP Token page, select the radio option **I use a QCode-based Authenticator (Time-based)** or **I use Google Authenticator** (if this option exists). WebADM will then present a QR code (see Figure 21).

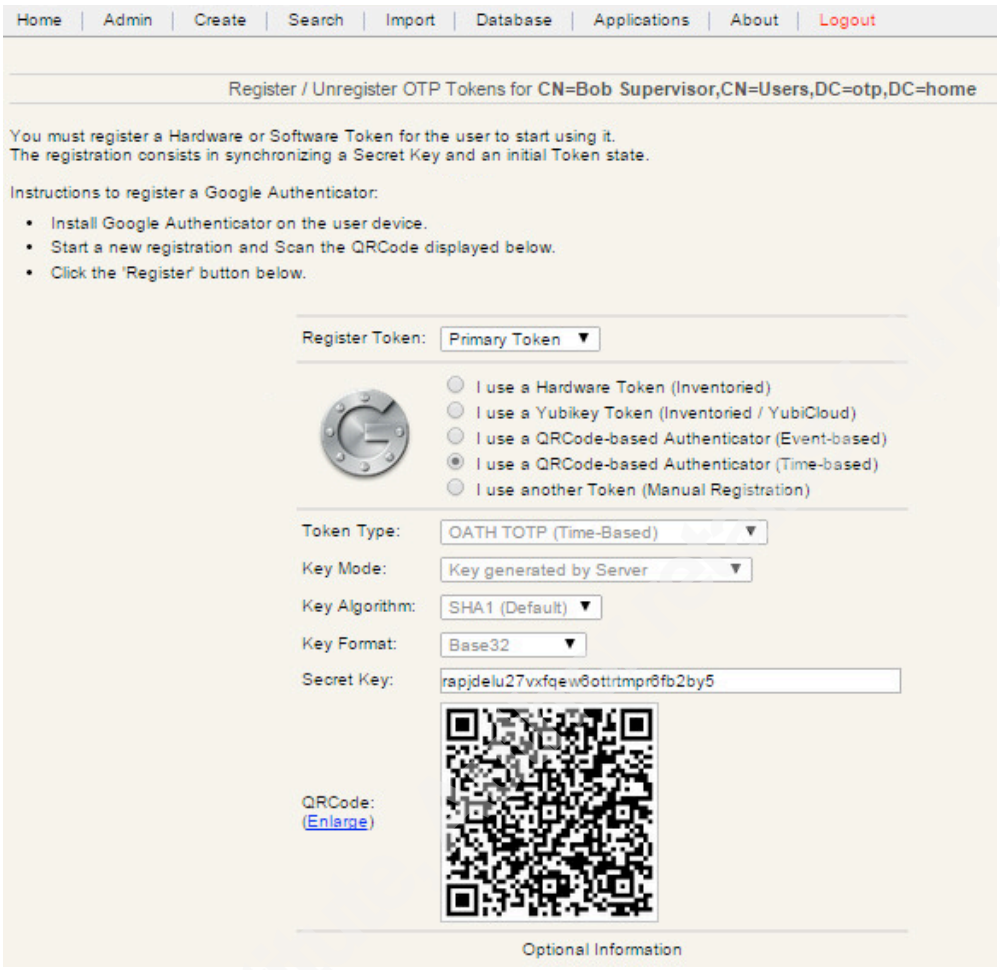

Figure 21 WebADM User OTP Token Registration

NOTE: A variety of token types for the user from this menu can be enabled, including the option to enable a pin code for the user to use with the TOTP. This would enable 2FA without the need to enter the user's LDAP password, similar to how RSA® multifactor tokens function. This feature is outside of the scope here.

6. Using the Google Authenticator application on a smartphone, press the "+" button and then select **Scan barcode** (see Figure 22).

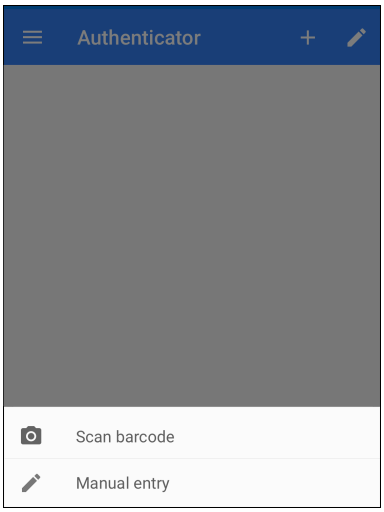

Figure 22 Adding a Token to Google Authenticator

7. After scanning the barcode, Google Authenticator should now display the TOTP token (see Figure 23) for the user "bob" (Bob Supervisor). Click **Register** on the WebADM user token registration page to finalize registering the TOTP for the user.

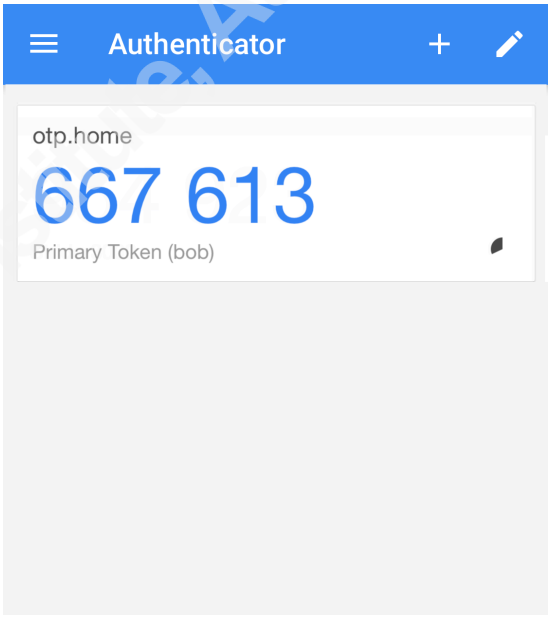

Figure 23 TOTP for User Bob Supervisor

8. To test the TOTP for the user, navigate back to the user account from the user directory navigation panel, select **OTP & U2F Authentication Server** from the list of Application Actions, and select **Test User Login**. This will bring you to a page where you can test the authentication for the user account using an LDAP name and password (see Figure 24). To test the user's

authentication, enter the LDAP password and current TOTP for the user (from the Google Authenticator application).

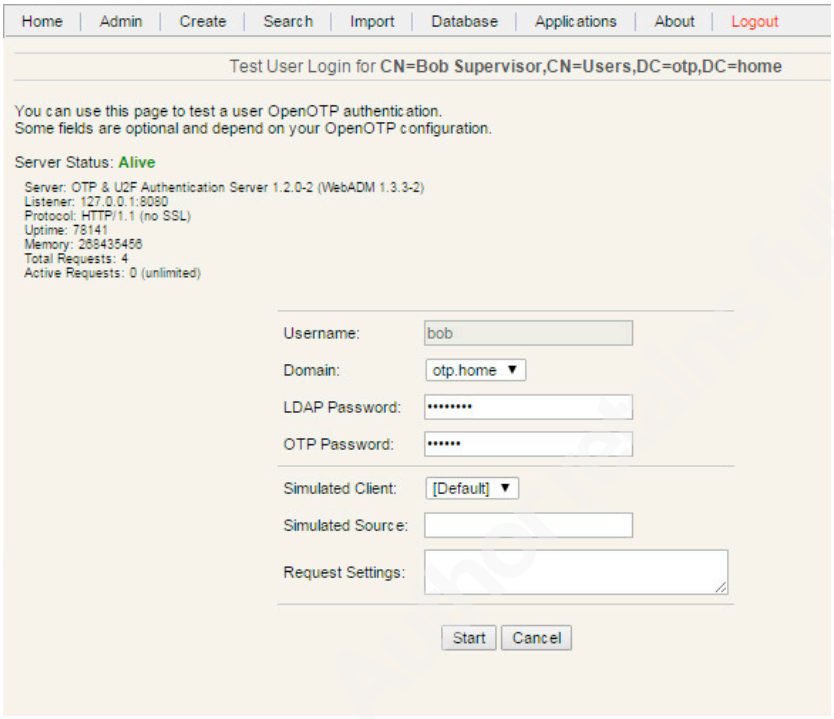

Figure 24 Testing the WebADM User TOTP and LDAP Authentication

9. WebADM should display an authentication success screen. If the TOTP and LDAP authentication fail to successfully authenticate, see the Troubleshooting section below for common troubleshooting steps.

10. Repeat the steps 3-9 for the Ted Technician and Ada Engineer users. Now, three different TOTP tokens on Google Authenticator application (see Figure 25) should be present.

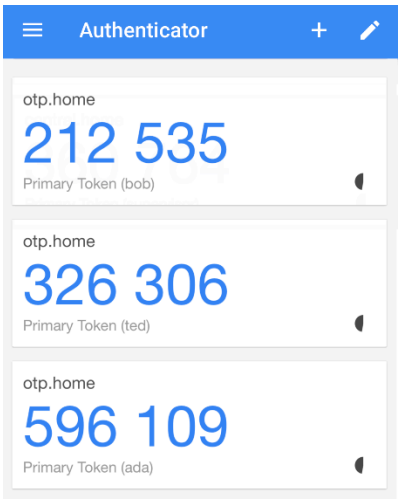

Figure 25 Bob, Ted, and Ada TOTP Tokens

NOTE: You can have a user self-configure their own TOTP using Google Authenticator by activating other WebADM services that are available. For example, the User Self-Service Desk application allows users to configure their own personal information (phone numbers, email addresses, etc.), LDAP password, and OTP tokens without requiring direct guidance from an administrator.

3.4.1. Troubleshooting

If you have problems authenticating using the test users' LDAP and TOTP password combinations, please see the following tips:

- Ensure the LDAP password for the user is correct.
- Ensure that you have successfully registered the TOTP token before attempting to use it.
- Ensure the RCDevs appliance is getting accurate time via NTP.
- Turn on Syslog logging in /opt/webadm/conf/webadm.conf and use the command "**tail –f /var/log/messages**" to see messages sent by the WebADM service in real time.
- You may also view logs directly from the WebADM service by using the command **"tail /opt/webadm/logs/soapd.log"** (see Figure 26).

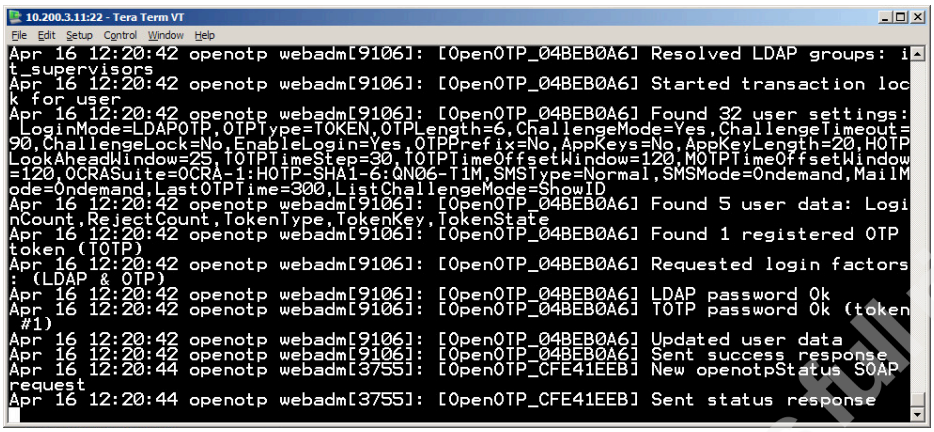

Figure 26 WebADM soapd.log File

3.5. Testing 2FA with OpenOTP RADIUS Bridge

Here we will test 2FA for one or more test users using the OpenOTP RADIUS Bridge service. Please ensure you have 2FA using LDAP password and Google Authenticator TOTP working successfully in the previous section before attempting to perform the steps in this section.

1. From the RCDevs appliance command-line, navigate to **/opt/radiusd/conf**. Use a text editor to open the file **clients.conf** (see Figure 27). This file specifies what authentication clients are allowed to access the OpenOTP RADIUS Bridge. By default, it allows all authentication clients to connect using a RADIUS secret of "testing123". The author changed this secret to **Asdf123$**.

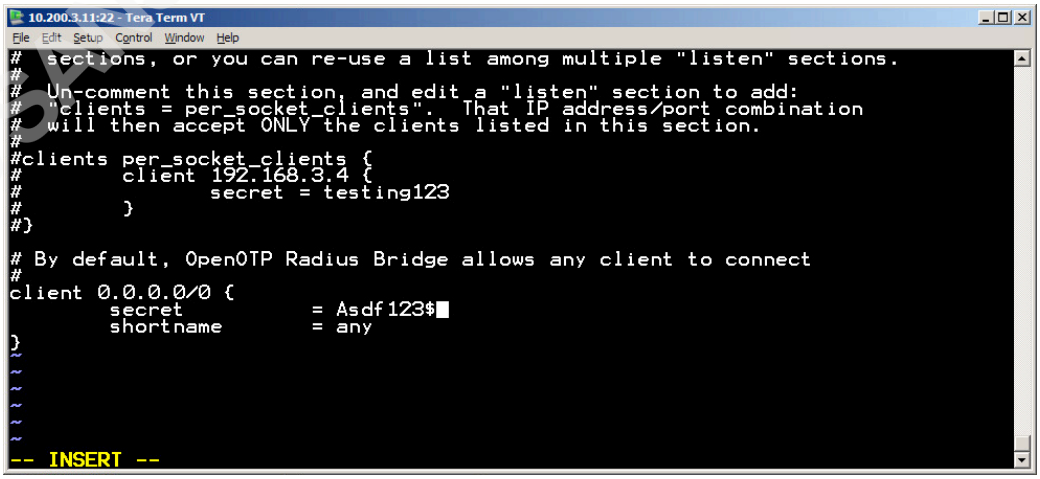

Figure 27 RADIUS Bridge clients.conf File

2. Finish editing the file and saving the changes, and then restart the radius service using the command **service radiusd restart**.

3. Next, navigate to **/opt/radiusd/bin/**. In this directory is a file called **radtest** that will allow us to test authenticating to the OpenOTP RADIUS Bridge. To use radtest, you may call the program using the arguments:

radtest <user> <IP address of RADIUS server>:<RADIUS server port> <RADIUS shared secret>

In the case of the author, the command to test the Ada Engineer user's authentication is:

./radtest ada 127.0.0.1:1812 Asdf123\$

Note that on systems using the bash shell, the '$' symbol is special, and therefore must be escaped using the backslash '\' symbol (see Figure 28).

Figure 28 Using radtest

4. At the prompts, enter the password and then the TOTP token for the user Ada Engineer. If the password and TOTP token for Ada Engineer are good, then you should see a success message (see Figure 29).

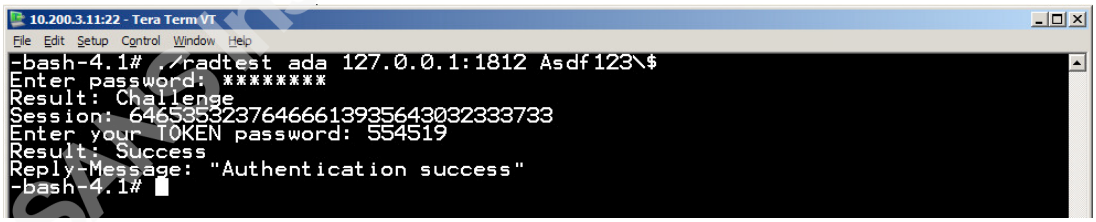

Figure 29 RADIUS Authentication Success Message

5. Repeat steps 3-4 for Bob Supervisor and Ted Technician. You should receive success messages for each. If not, please see the troubleshooting steps and the end of this section.

3.5.1. Troubleshooting

If you have problems authenticating to the OpenOTP RADIUS Bridge using the radtest client, please see the following tips:

- Ensure the LDAP password and TOTP token for the user are correct, and that you have registered the TOTP token for the user.
- Ensure the RCDevs appliance is getting accurate time via NTP.
- You may download and install tcpdump, then using the command "**tcpdump –i eth0 port 1812 –vvv**" to show incoming RADIUS packet requests and responses in real time.
- You may view **/opt/webadm/logs/soapd.log** for OpenOTP Authentication Server debug information, or **/opt/radiusd/logs/radiusd.log** (see Figure 30) for **/opt/radiusd/logs/requests.log** for or debug information from the radiusd service.

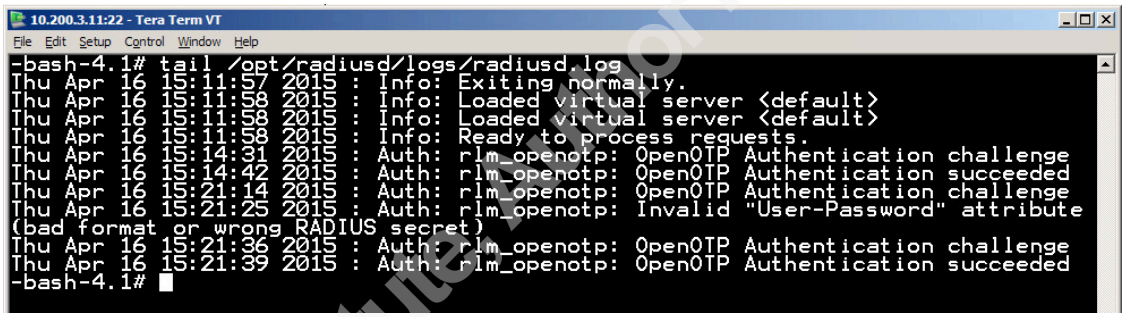

Figure 30 radiusd.log File

4. Integrating an OpenVPN Access Server with OpenOTP

To configure an OpenVPN Access Server (AS) authentication client that with RADIUS protocol, please follow the steps outlined in this section.

NOTE: the author uses an OpenVPN AS server virtual instance downloaded from the openvpn official website. The IP address of the author's OpenVPN AS appliance is 10.200.3.12.

1. Navigate to your OpenVPN AS instance web graphical user interface (GUI) and select User Authentication. From this window, select RADIUS as the user authentication method (see Figure 31). Be sure to click Save Settings.

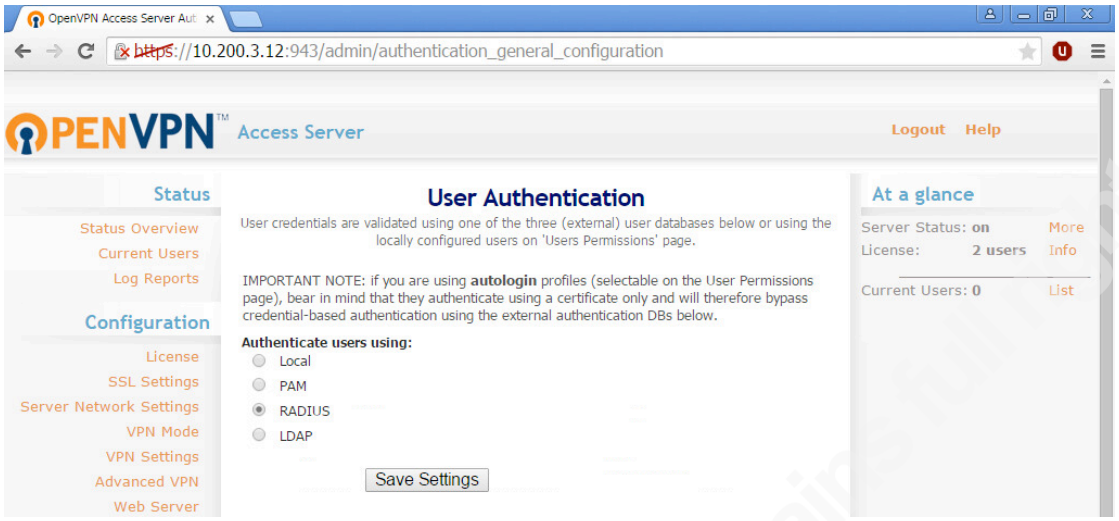

Figure 31 OpenVPN AS User Authentication Selection

2. Next, navigate to **Authentication → RADIUS** and enter the IP address, port, and shared secret for the OpenOTP RADIUS Bridge. Select **PAP** as the RADIUS authentication method like in Figure 32 below. Be sure to click **Save Settings** after entering your configuration settings.

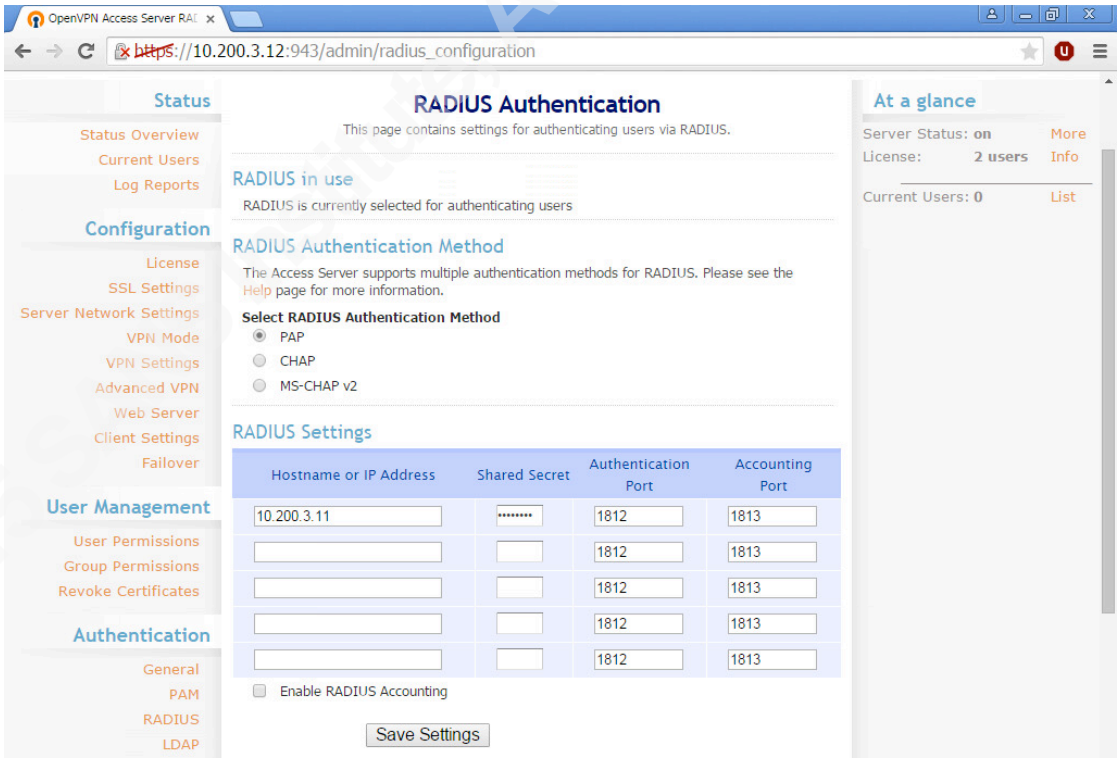

Figure 32 OpenVPN AS RADIUS Settings

3. You may test access by attempting to authenticate as one of our three test users. Log out of the web GUI, then attempt to login as the Bob Supervisor

(see Figure 33) test user (see Table 3 for our list of test users). Click **Go** to attempt the log in.

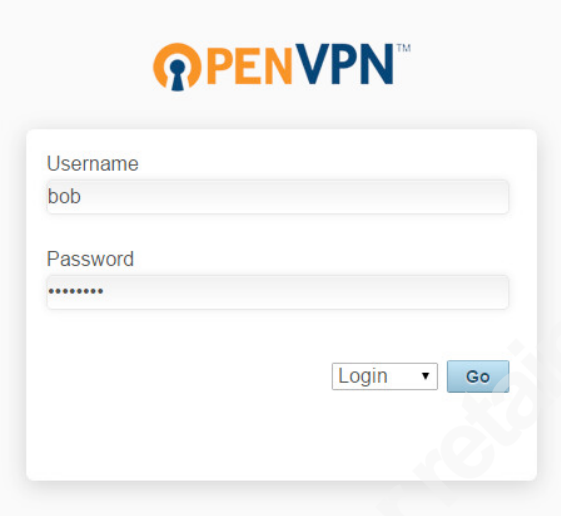

Figure 33 OpenVPN AS Web GUI Login

4. If you have successfully configured the RADIUS settings in the OpenVPN AS interface, the web page should greet you with a challenge screen asking the user to enter the token (see Figure 34).

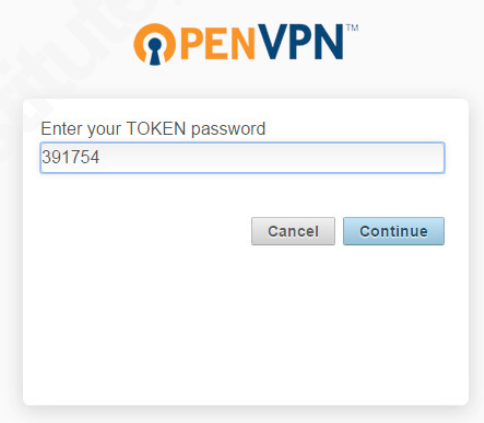

Figure 34 OpenVPN AS Challenge Screen

5. Enter the TOTP code for the Bob user as seen on the Google Authenticator application (see Figure 35), and click **Continue**.

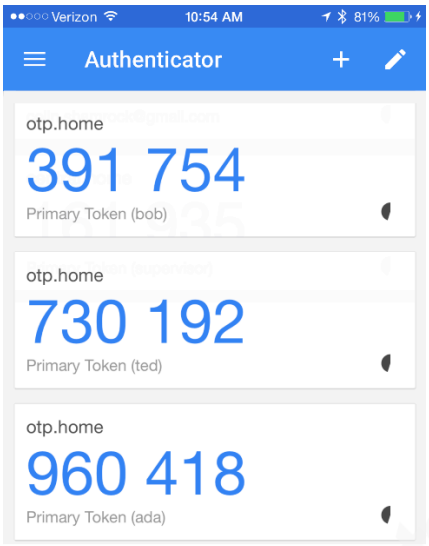

Figure 35 Google Authenticator Token for Bob

6. If the Bob Supervisor user has logged in successfully, then OpenVPN AS should greet you with the login selections page (see Figure 36).

Figure 36 OpenVPN AS Successful Login Screen

7. You may perform this same challenge-based authentication process for the OpenVPN AS login using the official OpenVPN AS client application. When authenticating to the OpenVPN AS using the AS client application, enter username and AD password as usual (see Figure 37).

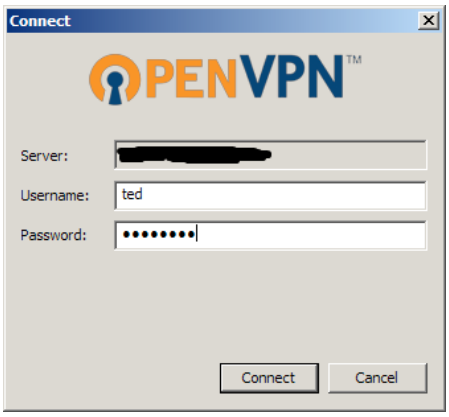

Figure 37 OpenVPN AS Client Application

8. You should receive a challenge message back from the OpenVPN AS instance (see Figure 38). Here the author entered the TOTP code off the Google Authenticator application for user Ted.

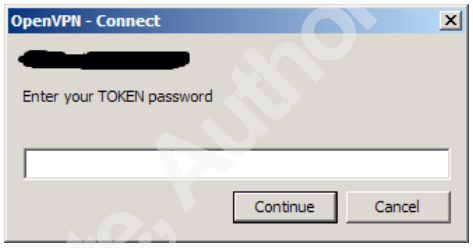

Figure 38 OpenVPN AS Client Challenge

NOTE: The official OpenVPN AS client does support challenge mode. However, the unofficial OpenVPN client (currently 2.3.6 as of April 20, 2015) does not support challenge mode, and will fail the authentication session if it receives a challenge from the OpenVPN AS instance.

9. Similarly, the OpenVPN Connect application for smartphones should support challenge mode. Once you have imported the OpenVPN AS connection profile into the OpenVPN Connect application, you will be able to authenticate using username and AD password (see Figure 39).

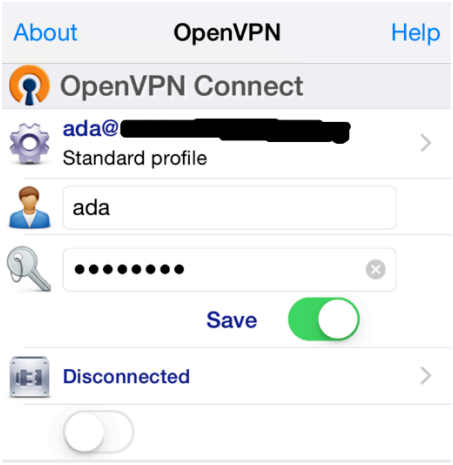

Figure 39 OpenVPN Smartphone Application

10. The OpenVPN AS instance will send back a challenge from the OpenOTP Authentication Server. The OpenVPN application will be display the challenge on the application screen (see Figure 40).

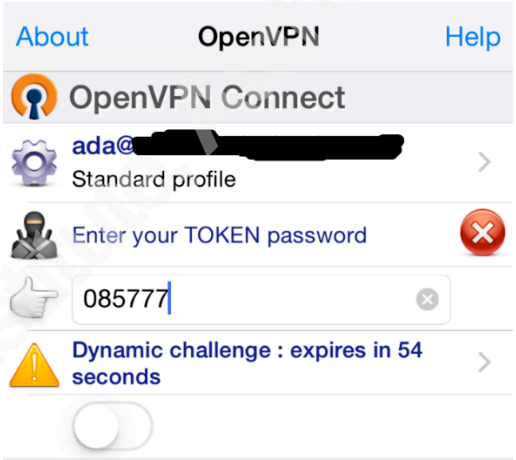

Figure 40 OpenVPN Smartphone App Challenge

NOTE: OpenVPN AS does support additional user authorizations using RADIUS attributes [4].

4.1.1. Troubleshooting

If you are having trouble authenticating to OpenOTP via the OpenVPN AS instance, please see following troubleshooting steps.

- Ensure the LDAP password and TOTP token for the user are correct, and that you have registered the TOTP token for the user.
- Ensure the RCDevs appliance is getting accurate time via NTP.

- You may view **/opt/webadm/logs/soapd.log** for OpenOTP Authentication Server debug information.
- You may view **/opt/radiusd/logs/radiusd.log** or **/opt/radiusd/logs/requests.log** for or debug information from the radiusd service.
- You may download and install tcpdump, then using the command "**tcpdump –i eth0 port 1812 –vvv**" to show incoming RADIUS packet requests and responses in real time.
- If you are having trouble authenticating to the OpenVPN AS instance, please ensure that you are using an OpenVPN AS client that supports challenge mode.
- If you are having problems remaining connected to the OpenVPN AS instance, please ensure that you have configured the "**reneg-sec 0**" parameter on both the OpenVPN server and client profiles. This will prevent the OpenVPN server from attempting to renegotiate the, which forces the user to re-authenticate to the OpenOTP Authentication Server using a username/password + TOTP code.

5. Conclusion

By now the reader should have an understanding of the effort involved with integrating 2FA into an existing user directory, and the basics for configuring various authentication client devices to interact with the OpenOTP Authentication Server. Administrators can configure most any device that supports RADIUS for user-based authentication purposes to allow users to use their AD username/password with Google Authenticator token. Solutions such as OpenOTP are inexpensive, and are easy to configure and maintain (especially for smaller organizations).

Author Name, email@address

References

Davis, J. (2015, June). *Two Factor Auth (2FA)*. Retrieved from https://twofactorauth.org

Davis, J. (2015, June). *Two Factor Auth (2FA) Providers*. Retrieved from Two Factor Auth: https://twofactorauth.org/providers/

IETF. (2005, December). *HOTP: An HMAC-Based One-Time Password Algorithm*. Retrieved from RFC 4226: http://tools.ietf.org/html/rfc4226

IETF. (2011, June). *OCRA: OATH Challenge-Response Algorithm*. Retrieved from RFC 6287: http://tools.ietf.org/html/rfc6287

IETF. (2011, May). *TOTP: Time-Based One-Time Password Algorithm*. Retrieved from RFC 6238: http://tools.ietf.org/html/rfc6238

NERC. (2014). *CIP-005-5 — Cyber Security – Electronic Security Perimeter(s).* Retrieved from www.nerc.com: http://www.nerc.com/_layouts/PrintStandard.aspx?standardnumber=CIP-005-5&title=Cyber%20Security%20-%20Electronic%20Security%20Perimeter(s)

OpenVPN. (n.d.). *Documentation*. Retrieved April 20, 2015, from http://openvpn.net/index.php/access-server/docs/admin-guides/411-access-server-post-auth-script.html

Sophos. (2014, January). *The Power of two - All you need to know about two-factor authentication*. Retrieved from Naked Security: https://nakedsecurity.sophos.com/2014/01/31/the-power-of-two-all-you-need-to-know-about-2fa/

Sophos. (2014, November). *Two-factor authentication: Understanding the options*. Retrieved from Naked Security: https://nakedsecurity.sophos.com/2014/11/14/understanding-the-options-2fa/

Verizon. (2015). *Data Breach Investigations Report.* Verizon Enterprise Solutions. Retrieved from www.verizonenterprise.com/BDIR/

Author Name, email@address

Appendix A
How OpenOTP Works

OpenOTP requires three pieces to fully authenticate a user using an OTP and password: the OpenOTP RADIUS Bridge, the OpenOTP Authentication Server, and a User Directory Service.

RADIUS Bridge: Remote Authentication Dial-In User Service (RADIUS) protocol is ubiquitous and interoperable with most all authentication clients and systems. The RADIUS protocol is simple, easy to configure, and extremely flexible. OpenOTP's RADIUS Bridge, relies on open-source FreeRADIUS 2, and is essentially a plug-in that allows most authentication clients to take full advantage of OpenOTP's 2FA capabilities.

OpenOTP Authentication Server: the OpenOTP Authentication Server ensures the authenticity of the one-time passcodes and other passwords the user enters. OpenOTP Authentication Server also handles user and group membership, and acts as the "brain" that decides how to authenticated users and what authorizations on the authentication client they may have.

User Directory Service: this is the service that stores user information. The most common user directory service is Microsoft Active Directory® (AD), which makes use of Lightweight Directory Access Protocol (LDAP) as its interface language. AD stores users along with group memberships and other identifying attributes. OpenOTP stores OTP information about itself and users/groups in a directory service (in an encrypted form). OpenOTP comes with its own open-source OpenLDAP server. An administrator can configure OpenOTP to connect to non-AD based LDAP servers, such as OpenLDAP-based directory services.

OpenOTP requires WebADM Web-Based Directory Administrator for configuring OpenOTP Authentication Server. Most configuration steps listed later will involve manipulating the OpenOTP environment through WebADM.

In addition to RADIUS, OpenOTP supports SOAP authentication requests directly from web authentication clients. This allows additional flexibility for administrators to integrate 2FA into web applications and internal business websites.

Author Name, email@address

OpenOTP also has a number of WebApps that administrators may be use for easier management of user data, including a user help desk, a token services desk, and so on. These services help ease the management overhead by allowing users to configure their own tokens and reset their own passwords without direct intervention.

Here, we will be using Microsoft® Active Directory® as the user directory service. OpenOTP Authentication Server uses the LDAP protocol to perform lookup and administrative functions for users/groups in AD.

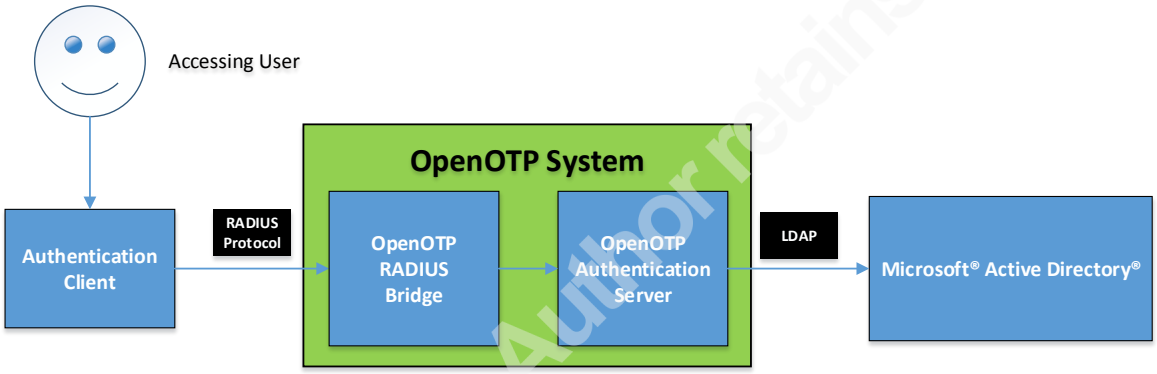

Figure 41 OpenOTP Authentication Architecture with Microsoft AD

Appendix B
OpenOTP Virtual Image Initial Configuration for VMware ESXi

This appendix expands the initial configuration steps for the RCDevs virtual appliance, including configuring the server hostname, IP address, changing the root password, and ensuring the virtual appliance can receive accurate time via Network Time Protocol requires (NTP; critical for OATH TOTP accuracy).

5. Download the OpenOTP Virtual Appliance (OpenLDAP – OVF) from https://www.rcdevs.com/downloads/index.php?id=VMWare+Appliances#

Figure 42 RCDevs Virtual Appliance Download

6. On the ESXi, click File → Deploy OVF Template, and select the unzipped "RCVM-OpenLDAP\RCVM-OpenLDAP.ovf" that was just downloaded and click **Next**. At this point you may receive a warning that Red Hat Enterprise Linux 6 is not supported. Click **Yes** to continue.
7. Choose the name for the virtual appliance, disk format, and networks for the RCDevs virtual appliance. Please ensure your network you select for the RCDevs image is reachable by the Windows Server and your authentication clients.
8. Finish the deploying the RCDevs virtual appliance and power on the system (default Memory and CPU settings should be fine for testing).
9. During the initial boot, the Linux system will prompt you to input the server Fully Qualified Domain Name (FQDN), to enter an organizational name, to enable WebADM to be started automatically, to register the WebADM logrotate script, and to generate a WebADM secret key (the key encrypts WebADM information in the user directory service). The author used

openotp.otp.home as the FQDN, **Self** as the organizational name, and "yes" to all else (see Figure 43).

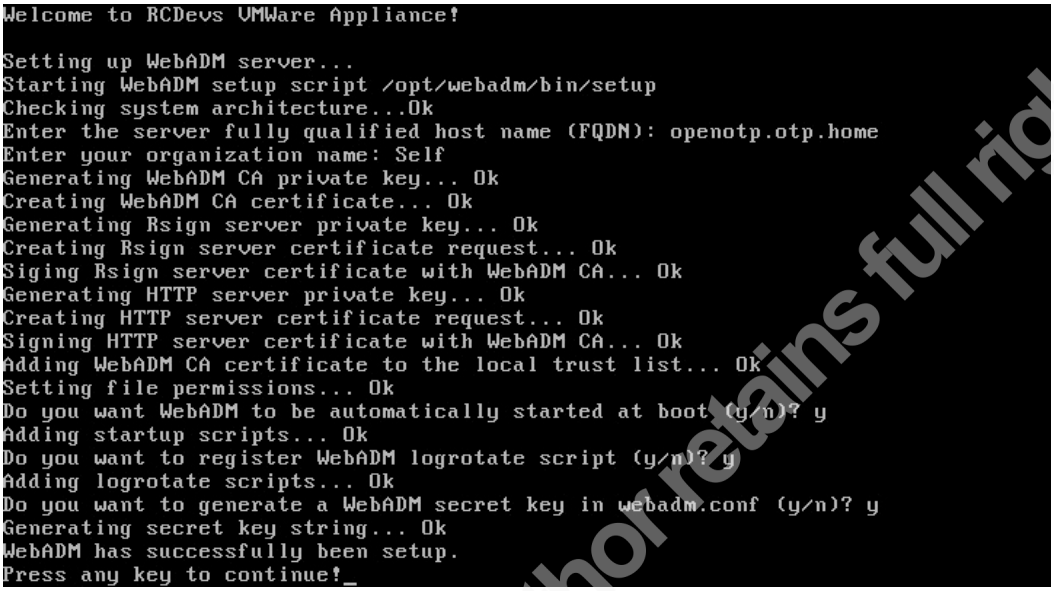

Figure 43 RCDevs Appliance Initial Configuration

10. If there is a DHCP server on the network the RCDevs appliance connects to, you should see information about the IP address of your network. Additionally, the RCDevs appliance will list the default root account credentials (**root** / **password**, see Figure 44). If there is no IP address assigned to the RCDevs appliance, you may assign one manually.

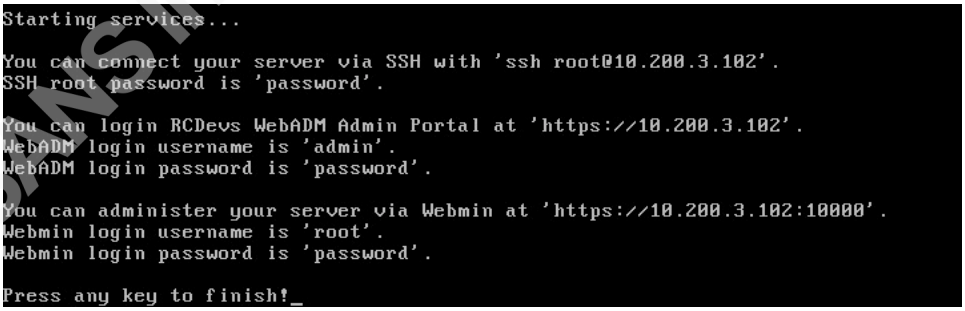

Figure 44 RCDevs Appliance Initial IP and User Information

11. From a web browser, navigate to the IP address of the RCDevs appliance to log in to the WebADM configuration platform using HTTPS on port 10000 (see Figure 45) to finish the initial configuration (https://10.200.3.102:10000 in the case of the author). You may use the default root credentials to log in (**root** / **password**).

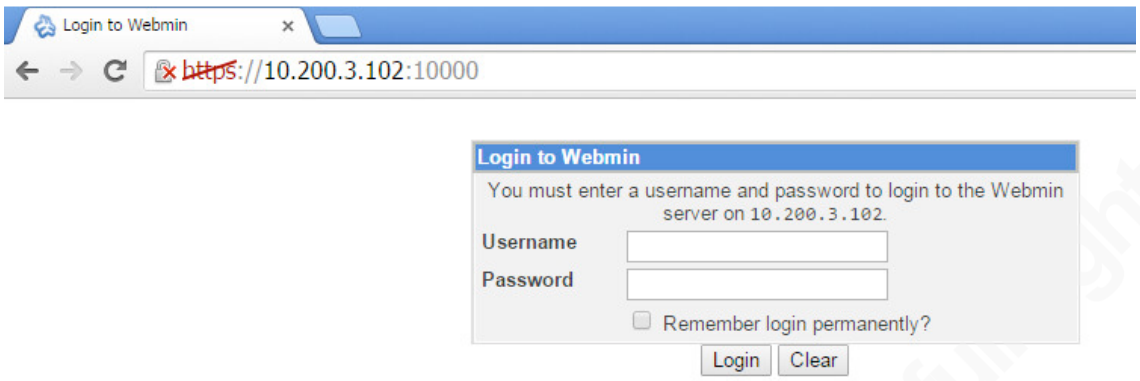

Figure 45 Login to WebADM Platform

12. From the WebADM dashboard, we will set the root password, the network IP address, and the time service on the server. To change the root password, navigate to **System → Change Passwords → root** and set a new root password. The author uses root password **Asdf123$** for testing.

13. To change the appliance IP address and add a default gateway, navigate to **Networking → Network Configuration → Network Interfaces**. The author uses a static IPv4 configuration with an IP address of **10.200.3.11** and netmask **255.255.255.0** (see Figure 46), and a default gateway of **10.200.3.1.**

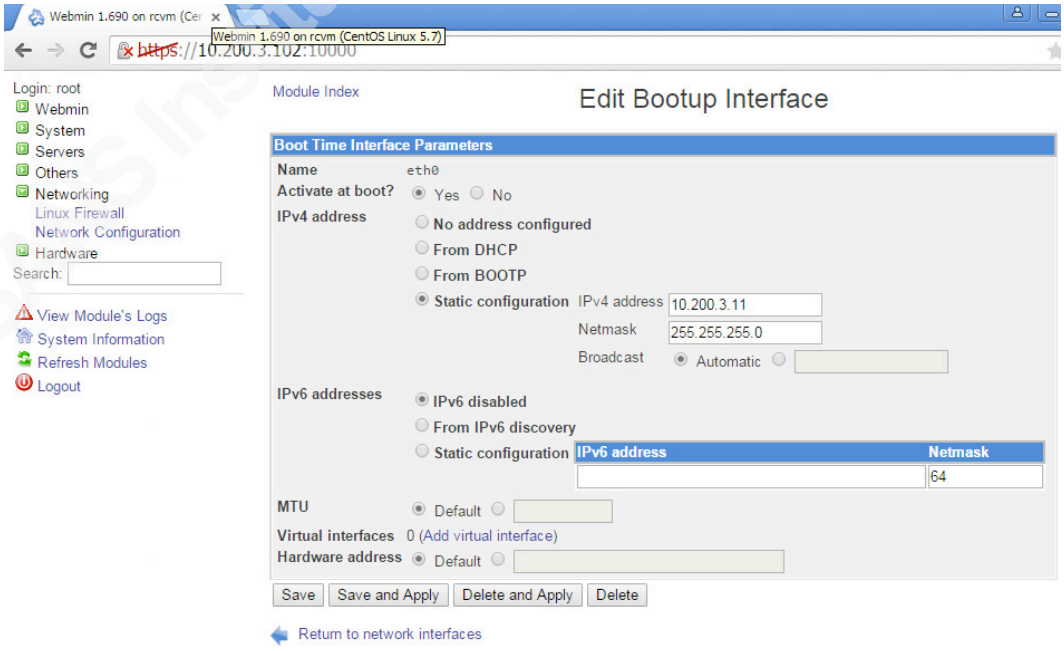

Figure 46 WebADM Network Configuration

14. Next, you may add a DNS server (either internal or external) by navigating to **Networking → Network Configuration → Hostname and DNS Client**. The author uses the hostname "**openotp.otp.home**", and a DNS server of **10.200.3.10** (the Windows Domain Server), and search domain of **otp.home**.

15. To set the local time zone and set NTP settings, navigate to **Hardware → System Time**. You may set the time zone from the **Change time zone** tab (Pacific in the author's case) and set the NTP settings from **Time server sync**. The author recommends synchronizing when the appliance starts, and also synchronizing on schedule (see Figure 47). Please note that you may need to reboot the appliance before you are able to get accurate time via NTP.

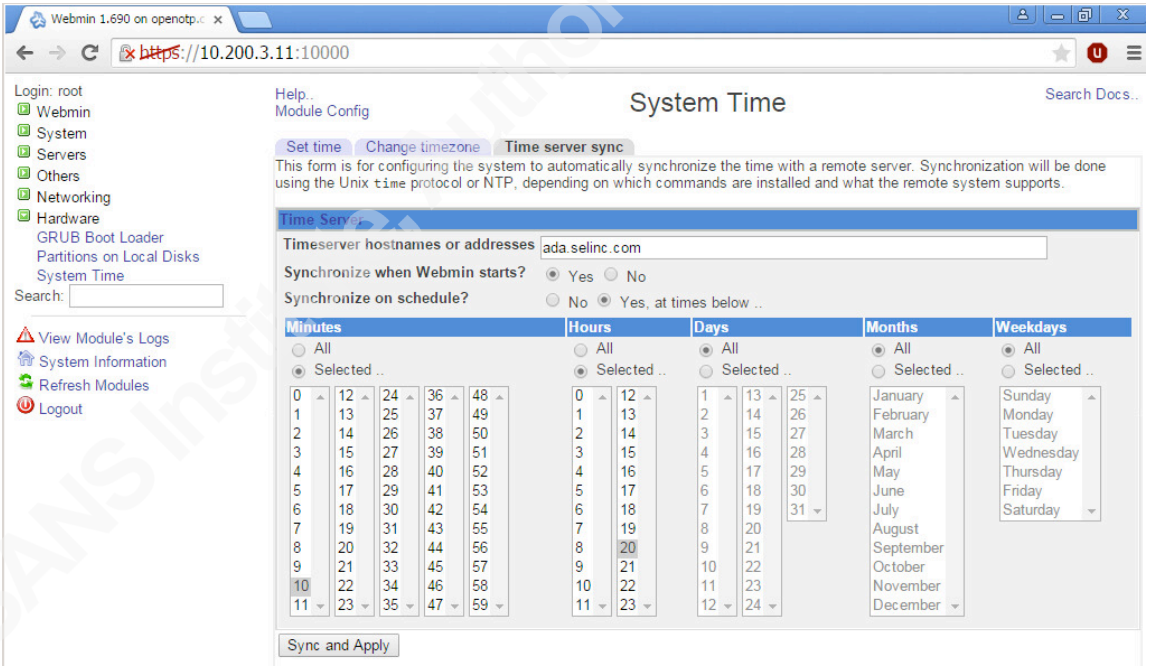

Figure 47 WebADM Time Sync Settings

16. After initial configuration, please ensure you can ping you Windows AD server at its hostname (in the author's case **adds.otp.home**) before continuing on to the next section (see Figure 48).

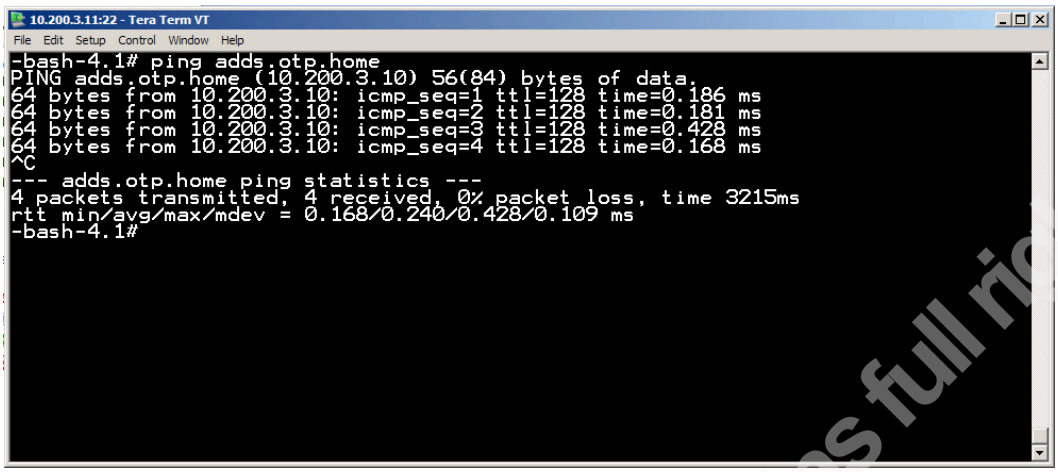

Figure 48 Pinging Windows AD from RCDevs Appliance

Author Name, email@address