RADIUS / AAA & Captive Portal Cloud Service™

Technical Overview
Cloudessa "AAA & Captive Portal Cloud Service"

Welcome to the Cloudessa® cloud based Authentication, Authorization, and Accounting (AAA) and Captive Portal solution platform.

Cloudessa supports building complex WiFi Captive Portals for hotspots such as retail, hospitality and guest access, as well authenticating enterprise users using 802.1X and RADIUS based protocols. It supports a variety of backend authentication sources, such as Google Apps, Active Directory, SAML providers, and social network sign-ins, including Facebook and Twitter, and Payment Processors such as PayPal.

Cloudessa is a 100% cloud service compatible with enterprise WiFi AP’s and Controllers from leading network hardware vendors.

Cloudessa is available either as a public cloud Service, or as a Virtual Appliance for installation in an enterprise or private data center, so you can deploy Cloudessa in the way that is appropriate for your business.

• Use the hosted Cloudessa Service in the public cloud, where you can take advantage of a shared multi-tenant infrastructure.

• Deploy Cloudessa as a Virtual Appliance, running in a private cloud or enterprise data center if you need to maintain service completely on-site and control service availability. Cloudessa VA runs on major private cloud platforms such as VMWare and Xen.

Why choose Cloudessa?

Driven by mobile workers and BYOD, the scale, complexity, and importance of enterprise WiFi and VPN networks is increasing dramatically.

A well-architected, multifaceted access security infrastructure is an essential element of every enterprise WiFi, VPN, and other remote access gateway deployment. This infrastructure typically must support the following functions:

• Authentication, to ensure that only authorized users gain access to the network
• Authorization, to configure the appropriate level of network resource access for a particular user or device for a particular session.
• Accounting, to document who access the network, when.
• Security, to prevent attacks on user credentials and data

In addition, these new WiFi requirements should ideally reuse existing user stores and integrate into the network’s existing access management systems and
architecture for securing VPN’s and other access gateways to ensure a consistent level of security regardless of how users are accessing your network.

WiFi hotspots provide unique set of business growth opportunities to engage consumers and guests of your business. A strong Captive Portal solution integrated into your business logic provides an opportunity to win new customers and keep new customers happy.

Cloudessa is the first cloud solution that enables you to both:

- Provide strong network access security for the employees and contractors of your business
- Grow your business by engaging your customers with advanced hotspot and captive portal solutions integrated with your business logic

Cloudessa enables you to achieve these goals while keeping the security of your network intact. Cloudessa supports the industry standard means of using separate WiFi SSIDs or network VLANs to separate your internal business network from customer engagement and hotspot network.

**Key Features**

The following are key features of Cloudessa discussed throughout this manual:

- Multiple Virtual RADIUS/802.1X servers, each running on a separate authentication and accounting port. You can create a Virtual RADIUS server with a single click of a mouse.
- Multiple Captive Portals, each running on a separate URLs. You can create a Virtual RADIUS server with a single click of a mouse.
- Captive Portal support based on industry standard UAM as well as Meraki EXCAP protocol and are compatible with major enterprise WiFi hardware such as Cisco, Meraki, Ruckus, Aruba, Motorola and others.
- For RADIUS/802.1X, a comprehensive variety of protocols are supported including PAP, CHAP, MS-CHAP, SIP, PEAP, EAP-TTLS, EAP-TLS and MAC-based authentication
- For Captive Portals, SAML authentication is supported, including such vendors as Ping Identity, Okta, OneLogin and Microsoft ADFS
- For Captive Portals, social network OAuth logins are supported, including Facebook, Twitter, and LinkedIn, as well as PayPal login for payment integration
- Accounting and Billing. Cloudessa includes build-in integration with PayPal, as well as a capability to add custom modules to integrate with other Payment Processors.
- Accounting Logs of user and admin actions
• Two-factor authentication using Google Authenticator
• Authentication against External user stores, including Active Directory, LDAP, SQL Databases, Google Apps, as well as customer-provided Web Service APIs
• JSON-based Web Services API
• Powerful Captive Portal building tools and widgets
• Three methods to use Google Apps for authentication: Captive Portal, PAP/EAP-TTLS and EAP-TLS with digital certificates.

The following figure illustrates how a multi-location enterprise can leverage the Cloudessa RADIUS functionality service in the public cloud to authenticate and authorize WiFi users and devices.

Figure 1: Example Deployment - Cloudessa RADIUS / AAA Cloud Service
Figure 2: Example Deployment - Cloudessa AAA & Captive Portal Cloud Service
Authentication Options

When assessing your WiFi and VPN network security requirements, it is important to examine what is the right level of security for your deployment, and how do you want to enforce the access security.

Cloudessa RADIUS provides the flexibility to deploy both WPA2 / 802.1X compliant or Captive Portal browser based access security.

Best practice for WiFi and VPN access to enterprise LAN applications mandates the use of WiFi Protected Access 2 Enterprise (WPA2) and 802.1X-based security; in addition, WPA2 and 802.1X are considered essential for securing WiFi access in healthcare (HIPAA), financial services (SOX), and other regulated environments.

If the primary use of the WiFi network is to access cloud or external resources, (for instance in a hotspot or for student / customer / guest internet access) or if a users session will be protected via a VPN tunnel, and there is little risk of sensitive data being compromised, then a browser based login via a Captive Portal is a viable option.

WPA 2 / 802.1X or Captive Portal

With WPA 2 / 802.1X, authentication happens before a user is granted an IP address and allowed on the network, this protects against attacks at upper layers by denying access before a rogue user ever gets on the network. WiFi networks requiring a high level of access security and most VPN networks use WPA 2 / 802.1X based access security.

WPA 2 / 802.1X works at Layer 2, the data link layer. In this case, the wireless client is authenticated, the encryption key is derived and the Layer 2 wireless connection between the client and the access point is encrypted. WPA2 supports Extensible Authentication Protocol (EAP) based authentication to prevent access until user authentication is completed.

The 802.1X protocol applies to wired and wireless networks. In a wireless network, the 802.1X authentication occurs after the client (end user) has associated to an access point using an 802.11 association method. Wired networks use 802.1X by connecting to a port on an 802.1X enabled switch.

Captive Portal provides a browser-based mechanism for user to login to the network. With Captive Portal, unauthenticated users attempting to access the
network are redirected to a Captive Portal web page. Users access to network resources is restricted until they are authenticated via a browser-based login.

Captive Portal is an application-level authentication used primarily with WiFi for hotspot and visitor / guest access networks. With Captive Portal, the user does obtain an IP address on the network prior to authentication; however, their network usage is restricted until they are authenticated via a browser based login.

**Captive Portal authenticates users at Layer 3, the network layer.** In this case the encryption is typically done at the level of the browser using the HTTPS protocol. Captive Portal authentication is often used in conjunction with a layer 3 VPN, such as an IPSec or SSL VPN, that is used to encrypt the entire layer 3 traffic.

The decision to use WPA 2 / 802.1X or Captive Portal based access security depends on your access network infrastructure and security risk profile. Organizations who's employees will be using the WLAN or VPN to access corporate applications and resources and cannot risk their network or data being compromised should consider the more secure WPA 2 / 802.1X Layer 2 security approach.

If the primary use of the WiFi network is to access cloud or external resources, for instance in a hotspot or for customer / guest access, then Captive Portal Layer 3 security is an appropriate option.

**The Role of RADIUS and AAA**

Regardless of which method you choose for enforcing access security on your WiFi AP’s, VPN’s, or other access gateways, authenticating users to a network through client based WPA2 / 802.1X or browser based Captive Portal, Cloudessa RADIUS server provides advanced capabilities for both.

- The RADIUS server orchestrates and manages the interaction between a number of different network elements that need to work collaboratively to manage and secure WiFi Access Point’s and Controllers (AP’s), VPN’s, and other access gateways.

- A centralized RADIUS server receives authentication requests from the WiFi AP’s, controllers, VPN servers, or other access gateway.

- User credentials are then processed against a designated user store, typically Active Directory (AD), or an LDAP or SQL database.

- If a cloud user store such as Google Apps™, SAML or social network is used, Cloudessa RADIUS will create and delete the corresponding RADIUS credentials on the fly.

- Authentication is accepted or rejected based on the validity of the provided
user account credentials.

• When returning the access accept / reject message to the gateway, the RADIUS server also returns the parameters for the user authorization to network resources. The Authorizations are returned via standard and vendor specific RADIUS attributes, for each user and session, based on which group or groups the user is an authenticated member of (based on the users group assignments in AD, Google Apps or other user store)

• The role of the RADIUS server is essential. Not only does it authenticates the user, but it also communicates back to the gateway WiFi AP or VPN (via RADIUS attributes), the parameters for how that gateway should be configured for that particular user, for that particular session, based on what network group (as defined in AD or Google Apps or other user store) that the user is a member of. Such parameters can include assigning users to particular VLAN’s, setting bandwidth allocation, and dynamically configuring any other configurable policy element of your access gateway.

• RADIUS accounting logs are generated and stored to detail describing the user and the device accessing the network. RADIUS accounting logs can be important for documenting who was on the network, when; and for proving accountability and security compliance within regulated environments such as healthcare, financial services and public access networks.

WiFi access security is dependent on the interoperability between a number of different network components:

• User Device, typically a laptop or smart device running "client" or "supplicant" software or a browser;

• WiFi AP, WiFi Controller, VPN, Firewall or other Access Gateway - The Access Gateway is the access security enforcement point and is the "Authenticator" or "RADIUS Client" that initiates and sends the RADIUS authentication request to the RADIUS server;

• RADIUS Server – IETF Standards based server that handles the authentication, authorization, and accounting for user access;

• User Store – Active Directory, LDAP or SQL database, Google Apps, or other user store where user credentials and user group assignments are stored.

All of these network components must be configured and interoperable to enforce access security.
User Credential Stores

The following user stores and authentication sources are supported:

- Active Directory, LDAP, SQL databases,
- Google Apps
- SAML authentication, for instance Ping Identity, OneLogin, Okta and ADFS
- Social network OAuth-based logins, such as Facebook, Twitter LinkedIn, PayPal.
- Cloudessa internal native user store
- Customer-owned webservice APIs. Examples include hospitality, recreation, health-care and co-working spaces. In this case Cloudessa will call the external webservice API during authentication

RADIUS and 802.1X Authentication Protocols

Cloudessa supports a comprehensive set of RADIUS and 802.1X authentication protocols. All of these protocols include a shared secret between the RADIUS client and the RADIUS server. Typically RADIUS clients are WiFi Access Points or Controllers, VPN's or firewall devices.

Older, non-802.1X compliant protocols include:

**Password Authentication Protocol (PAP)** - The user enters a username and a password. The password is encrypted using the RADIUS shared secret and then the username and the encrypted password are sent to the RADIUS server, the server verifies them against a user store. The password may be stored in the user store in plaintext or as a hashed value. If the verification is successful, Accept message is sent back to the RADIUS client. PAP is one of the oldest and mostly widely used protocols in wired networking. It is also used in wireless networks for Captive Portal authentication using web forms, and for the EAP-TTLS/PAP protocol suite.

**Challenge Handshake Authentication Protocol (CHAP)** - is more secure than PAP. With CHAP, the server sends a random “challenge” string to the client, along with the hostname. The client uses the hostname to determine the appropriate secret, combines it with the challenge and returns the information to the server. The server acknowledges the client, and permits access if the correct result is received. In such a way the password is never communicated over the network, improving security over PAP.

**MS-CHAP v1 and v2** – is a Microsoft version of CHAP. MS-CHAP is an option in the Microsoft implementation of Point to Point Tunneling Protocol (PPTP).
MAC Authentication Bypass – important protocol which uses the MAC address of a device as the username and the password. Although this protocol is not particularly secure it is widely used for low security environments, such as guest access. Typically this protocol is implemented by wired Layer 2 switches and Layer 2/3 gateways.

Digest is a widely used username/password protocol for Voice-over-IP systems.

MSISDN is a RADIUS protocol variation where Mobile Subscriber Integrated Services Digital Network-Number (MSISDN) is used as the authentication credential. This protocol is used by telecom RADIUS servers.

Newer, 802.1X compliant protocols are described below. For 802.1X the user client (supplicant), typically installed on a laptop or wireless device, authenticates to the RADIUS server through the Authenticator, such as Access Point or wired Ethernet switch. The Authenticator plays the role of the relaying party helping the Supplicant and the RADIUS server exchange messages. Once the authentication is complete, the RADIUS server sends Accept message to the Authenticator, and the user is permitted to use the network.

802.1X protocols typically include a combination of a secure tunnel, and then the inner authentication protocol which is used over the secure tunnel once the secure connection is established. The secure tunnels include Microsoft PEAP, TTLS and TLS. PAP, CHAP or MS-CHAP are typically used as inner authentication protocols.

PEAPv0 / MS-CHAPv2 – this protocol is the most widely supported Wi-Fi authentication protocol, it used Microsoft PEAP as secure tunnel and MS-CHAPv2 as the inner authentication protocol. It is supported by Microsoft, Apple, Android and Blackberry devices. The limitation of this protocol is that the password needs to be stored on the server side in plaintext and cannot be hashed. Another limitation is that this protocol does not work with external web services, such as Google Apps, which typically can verify the password, but will not give out the password.

EAP-TTLS/ PAP – this protocol is uses Microsoft TTLS as secure tunnel and PAP as the inner authentication protocol. The password can be stored in hashed form, one can also use this protocol to authenticate against external web services. This protocol is supported natively on Android, Linux and Windows 8. On Apple devices, it is switched off by default and needs to be enabled. On older versions of Windows third party software such as SecureW2 needs to be installed to enable the protocol. A typical price of this third party software is $20-$50 per laptop.

EAP-TTLS/ MSCHAPv2 – not frequently used combination of TTLS and MSCHAPv2

Cisco LEAP – Cisco proprietary protocol. Used in older Cisco hardware.
**EAP-MD5** – Older protocol, not frequently used.

**RADIUS attributes**

One of the main reasons for the ubiquitous use of RADIUS in access networks is the flexibility of the RADIUS attributes to enable the application of a consistent set of access security policies across different types of access gateways, from different vendors.

Standard RADIUS Attributes define how an access gateway is configured for a particular user session. RADIUS attributes carry specific authentication and authorization details.

For example, to initiate a user session, the access gateway sends Access-Request packets to a RADIUS server. The initial packet contains several attributes that identify the user, such as username, password and other identifiers.

If the authentication is successful, the server responds with the Access-Accept packet that contains attributes that define the user session, such as VLAN and bandwidth limits.

RADIUS is extensible. In addition to the standard RADIUS attributes, networking vendors incorporate specific RADIUS attributes to add new capabilities for communication with the RADIUS server.

These attributes are contained in a RADIUS dictionary file. Vendor-specific dictionary files contain a definition of RADIUS attributes that are used by each vendor.

With Cloudessa RADIUS, you can select the level where access is authorized: you can define attributes at the individual user level, at the group level, or at the Virtual RADIUS server level.

Cloudessa frequently updates the vendor-specific RADIUS dictionary options to ensure that the latest files are available.

**Captive Portal Authentication**

Cloudessa includes a powerful set of tools to build Captive Portals. These Captive Portals are web-based and hosted by Cloudessa. Cloudessa utilizes the UAM (Unified Access Method) standard and the Meraki EXCAP protocol to integrate with a wide
variety of WiFi hardware, including Cisco, Meraki, Ruckus, Motorola, Aruba and others.

Cloudessa Captive Portal can include a number of authentication options, in particular:

- Social network login using Facebook, Twitter, and LinkedIn.
- Google Apps authentication
- Self-registration
- Login with PayPal and the corresponding billing/payment capabilities
- SAML-based authentication utilizing Secure Assertion Markup Language. Examples of supported SAML providers are Ping Identity, Okta, Microsoft ADFS, and OneLogin.

GOOGLE APPS AUTHENTICATION

User Names & Passwords: EAP-TTLS Option

To authenticate users to the WiFi network using their Google Apps domain account user names and passwords requires using EAP-TTLS / PAP (Extensible Authentication Protocol with Tunneled Transport Layer Security / Password Authentication Protocol.

To securely pass the users account credentials from the user device to the network and over to Google for authentication and authorization, EAP-TTLS / PAP first authenticates the connection between the WiFi AP (the "Authenticator" or RADIUS Client) and the RADIUS server and sets up a trusted secure tunnel between the Authenticator and the RADIUS server.

EAP-TTLS then sets up a second “inner” encrypted tunnel for secure transport of the users credentials, so that the intermediaries to the authentication process (the AP and the RADIUS server) are only passing encrypted users credentials.

Within the secure “inner” tunnel, a second authentication protocol, PAP (Password Authentication Protocol), is used to transport the end users credentials. To authenticate a user using their Google Apps user name and password, EAP-TTLS must be the outer authentication, while PAP must be used as the inner authentication protocol.

To use EAP-TTLS / PAP requires the use of an 802.1X supplicant.

The following Operating Systems all include 802.1X supplicants and support EAP-TTLS and PAP: Apple, iOS version 3.1.3 and higher and MAC OS; Android v2.1 and
higher and Google Chrome OS (for Chromebooks); Microsoft Windows v8+ (note: Windows Mobile does not support EAP-TTLS; and Blackberry 6A+.

Administrators can automate user supplicant configuration through the use of profile creation tools (ie: iOS Profiles) and scripting.

Alternatively, SecureW2’s “JoinNow MultiOS” is a wireless security deployment platform that includes a client with support for a full range of Extensible Authentication protocols (EAP) including EAP-TTLS/PAP. See www.securew2.com.

Please visit www.cloudessa.com/support for detailed information about configuring the various supplicants for EAP-TTLS / PAP, profiling and scripting tips, and the latest information about other operating systems.

**Certificates - EAP-TLS Option**

In lieu of user names and passwords, Google Apps domain owners can opt to issue X509 certificates to their Google Apps users and use them with EAP-TLS protocol for user authentication.

EAP-Transport Layer Security (TLS) is used in certificate-based security environments, providing mutual authentication, negotiation of the encryption method, and encrypted key determination between the client and the authenticating server.

To enable the use of certificate credentials in a WPA 2 compliant manner, the signed certificate must first be in the certificate store on the mobile device, and then the user must present that certificate during the WiFi authentication process using a EAP-TLS supplicant.

Cloudessa provides a functionality to create and sign certificates, as well as to email certificate-installation links to users. The users install the certificates by simply clicking the link inside the email. During the EAP-TLS based authentication, the certificate is validated, and the email address of the certificate owner is checked against a listing of current Google Apps domain users maintained in the Cloudessa native database.

When a user is deleted in Google Apps, the user certificate is revoked.

In case when a mobile device is lost Cloudessa provides an interface to revoke the certificate installed on the lost device and generate a new certificate for the user.

**Cloudessa Certificate Creation Tool**
To facilitate the creation and distribution of Certificates signed by Google Apps, Cloudessa has created a Certificate Creation Utility, that administrators can use to easily create certificates on behalf of their Google Apps users.

The tool enables the importing of user names and email addresses, the generation of signed certificates, and it automates the process of then sending the certs to user via email for easy insertion into the certificate store on their device(s).

Captive Portal Option

With the Cloudessa Service, you can also authenticate Captive Portal users using Google Apps.

Please see the Captive Portal section of this manual for how to configure browser based logins against Google Apps.

For supplemental information regarding using Google Apps credentials to the WiFi network, please see the Support FAQ section on Cloudessa.com.

Two-Factor Authentication

Cloudessa supports 2-Factor Verification with the Google® Authenticator application for smartphones.

To authenticate against a Cloudessa RADIUS server, a user must possess not only a valid password, but also the PIN number generated by the Google Authenticator application.

Google Authenticator incorporates time based one-time passwords (TOTP) to enable an additional layer of security.

Cloudessa supports the Google Authenticator application to enable 2-step authentication.

The Authenticator application generates a temporary six digit PIN that users must enter to gain access to a network. A new PIN code is generated each 30 seconds, and the user must enter the current PIN (in addition to their standard password for two-factor authentication - for example, mypassword,123456.) Without the PIN code, a user cannot authenticate.

2-Step Verification is also available for administrative access via the Cloudessa Web UI. (What does this refer to -- need to be more explicit)
Authentication with the Google Authenticator application is only available with a paid Cloudessa account.

Google authentication is enabled on a per-user basis. Once Google Authenticator access is enabled for a particular user, RADIUS and Web UI access for the user requires the PIN code generated by Google Authenticator as well as the password (for 2-step authentication).

**Cloudessa Web Services API**

Cloudessa provides JSON-based Web Services API that can be used to programmatically utilize Cloudessa services.

Contact Cloudessa support for API documentation.
Configuring your Devices for Cloud Captive Portal services.

Configuring Meraki.

1. Log in to Meraki Dashboard
2. Go to Access control section. Edit existing SSID or create a new one.
Set the following parameters:

- **Association requirements**: Open (no encryption)
- **Splash page**: Sign-on with
- **Sign-on with**: <your RADIUS server>
- **RADIUS for splash page**: Enter Cloudessa Cloud Captive Portal parameters (RADIUS Server IP, RADIUS Authentication Port, RADIUS Shared Secret)
- **Captive portal strength**: Block all access until sign-on is complete
- **Walled garden**: Walled garden is enabled
- **Walled garden ranges**: Enter list of IP addresses

**NOTE:** You should look through all of your portal authentication providers and copy lists of IP addresses to your Meraki device walled garden.

3. Go to the Splash page section
Set the following parameters:
- Custom splash URL
- URL or provide a URL where users will be redirected: Enter Cloudessa Cloud Captive Portal parameters (Splash Page URL).

You Meraki configuration is finished.

Ruckus Configuration

1. Log in to Ruckus ZoneDirector
2. Go to Configure section and create new Authentication Servers

![Configuration Form]

Set the following parameters

- **Name**: <Specify portal name>
- **Type**: RADIUS
- **Auth Method**: PAP
- **Backup RADIUS**: Yes
- **Primary Server**
  - **IP Address**: RADIUS Primary Server IP
  - **Port**: RADIUS Authentication Port
  - **Shared Secret**: RADIUS Shared Secret
  - **Confirm Secret**: RADIUS Shared Secret
- **Secondary Server**
  - **IP Address**: RADIUS Secondary Server IP
  - **Port**: RADIUS Authentication Port
  - **Shared Secret**: RADIUS Shared Secret
  - **Confirm Secret**: RADIUS Shared Secret

3. Create new Accounting Servers
Set the following parameters

- **Name**: <Specify name>
- **Type**: RADIUS Accounting
- **Backup RADIUS**: Yes
- **Primary Server**
  - **IP Address**: RADIUS Primary Server IP
  - **Port**: RADIUS Accounting Port
  - **Shared Secret**: RADIUS Shared Secret
  - **Confirm Secret**: RADIUS Shared Secret
- **Secondary Server**
  - **IP Address**: RADIUS Secondary Server IP
  - **Port**: RADIUS Accounting Port
  - **Shared Secret**: RADIUS Shared Secret
  - **Confirm Secret**: RADIUS Shared Secret

4. Create new Hotspot Services
Set the following parameters

- **Name**: <Hotspot name>
- **WISPr Smart Client Support**: Enabled
- **Smart Client HTTP Secure**: HTTPS
- **Login Page**: Splash Page URL
- **Authentication Server**: Specify server created in section 3
- **Accounting Server**: Specify server created in section 4
- **Walled garden**: Mention all IP addresses you want to grant access to
<table>
<thead>
<tr>
<th>Order</th>
<th>Destination Address</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>devotbaycrypt.com</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>2</td>
<td>app.cloudessa.com</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>3</td>
<td>31.12.64.0/18</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>4</td>
<td>31.12.24.0/21</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>5</td>
<td>84.238.24.152/29</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>6</td>
<td>176.31.15.8/30</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>7</td>
<td>176.31.15.12/30</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>8</td>
<td>60.132.179.152/29</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>9</td>
<td>217.118.112.32/29</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>10</td>
<td>62.190.58.68/23</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>11</td>
<td>66.220.144.0/20</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>12</td>
<td>69.0.3.176.0/20</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>13</td>
<td>204.15.20.22/22</td>
<td>Edit Clone</td>
</tr>
<tr>
<td>14</td>
<td>faststatic.ukama.net</td>
<td>Edit Clone</td>
</tr>
</tbody>
</table>
NOTE: You should look through all of your portal authentication providers and copy lists of IP addresses to your Ruckus device walled garden.

5. Create new WLAN
Set the following parameters

- **Name/ESSID**: <Your SSID>
- **Type**: Hotspot Service (WISPr)
- **Authentication Options**
- **Method**: Open
- **Encryption Options**
- **Method**: None
- **Hotspot Services**: Specify server created in section 4

Your Ruckus configuration is finished.