How to Configure SSH

https://campus.barracuda.com/doc/96026684/

The SSH daemon listens on the management IP address on TCP port 22. Connect to the firewall with SSH when performing software updates or other special maintenance tasks. You can use both external SSH clients, or connect via the SSH tab in Barracuda Firewall Admin.

You can use an external SSH client to open an SSH connection to the Barracuda CloudGen Firewall. You can also use the terminal integrated in Barracuda Firewall Admin; direct access to the Barracuda CloudGen Firewall is provided via SSH version 2. To access the SSH terminal, click the **SSH** tab.

You can enable event notifications for SSH, as well as view the SSH log for information such as system access and remote command execution. In the **Select Log File** list on the **Logs** page, you can find the sshd log file in the **Box** directory.

Using an External SSH Client

If you prefer to use your own SSH client, configure Barracuda Firewall Admin with the path to the executable:

1. In Firewall Admin, click on the hamburger menu on the top left to expand the options tab.
2. Select **Settings**.
3. Expand the **Client Settings** section.
4. In the **External SSH Client** field, enter the command for the external SSH client. Use `%ip` and `%user` to dynamically insert the IP address and username. E.g., `C:\putty.exe %user@%ip`

Configure SSH

Step 1. Configure Basic Settings

1. Go to **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > SSH**.
2. Click **Lock**.
3. To configure the general settings for SSH, click **Basic Setup** from the **Configuration** menu in the left navigation pane.
4. On the **Basic Setup** page for SSH, you can configure the following settings in the **General Settings** section:
   - **Event for SSH** – Specifies if event notifications should be triggered when the system succeeds or fails to start up or shut down (Events Daemon Startup Failed/Succeeded

[2070/2071] and Daemon Shutdown Failed/Succeeded [2072/2073]).
You can select any of the following options:
- *Startup Failure*
- *Startup/Shutdown Failure*
- *Startup/Shutdown Failure + Startup Success*
- *Startup/Shutdown Failure + Startup/Shutdown Success*

You are not notified when sshd is killed manually or just dies unexpectedly. These settings only pertain to sshd behavior during controlled start or stop sequences.

- **Allow TCP Forwarding** – Specifies if TCP is enabled or disabled. This setting is only available in **Advanced View** mode.

  Disabling TCP forwarding does not improve security. You must also deny shell access to users because forwarders can be installed with the `ssh` command.
- **Login Timeout** – The maximum length of time in seconds that a user has to successfully log in before the server disconnects. The minimum time limit is 10 seconds. The default length of time is 90 seconds.
- **Permit Root Login** – Permits or prohibits SSH logins for the root user.

  If you prohibit SSH logins for the root user, the following configuration entities will not work: **Box Exec** tab and **Software Update** tab.
- **Check User Home** – (Only available in **Advanced View** mode) Specifies whether sshd should check file modes and ownership of the user's files and home directory before accepting login. This is normally desirable because novices sometimes accidentally leave their directories or files writable. The default is *yes*.

Step 2. Configure Settings for SSH Version 2

To configure settings that are specific to SSH version 2:

1. Click **Advanced Setup** from the **Configuration** menu in the left navigation pane.
2. You can configure the following settings:
   - **Client Alive Interval** – Timeout interval in seconds after which, if no data has been received from the client, sshd will send a message through the encrypted channel to request a response from the client. If set to 0, no message will be sent.

     The client alive mechanism is valuable when the client or server depend on knowing when a connection has become inactive.
   - **Max. Client Alive Messages** – Number of client alive messages that may be sent without sshd receiving any messages back from the client. If this threshold is reached while client alive messages are being sent, sshd will disconnect the client, terminating the session.
   - **Allow Compression** – Specifies if compression should be enabled or disabled for SSH clients.
   - **Force Key Authentication** – (Only available in **Advanced View** mode) Specifies if key usage is mandatory or optional for SSH clients.

     If key usage is mandatory for external SSH clients and you want to automate user logins, the private key of the client certificate on the Windows system must be generated in a UNIX-compatible format. For instructions on how to generate the required key, see the following "Handling Forced Key Authentication" section.

- **Secure FTP Support** – (Only available in **Advanced View** mode) Specifies if sshd implements the sftp subsystem. Secure FTP may be viewed as a more comfortable alternative to the simple scp command when trying to transfer bulk data to or from the box.
- **Use DSA Keys** – (Only available in **Advanced View** mode) To enhance security, other key types will be preferred over DSA keys. Disabling DSA keys increases security, but breaks the SSH communication with HA partners, Control Centers, or managed firewalls if the release version is below 7.1.0.

Step 3. Configure Brute Force Protection

Enable Brute Force Protection to let the the system watch login attempts of every account and block an account after a given number of consecutive failed logins. Brute Force Protection does not tell a possible attacker that their account got blocked.

1. On the **Advanced Setup** page, configure the following settings in the **Brute Force Protection Settings** section:
   - **Enable Brute Force Protection** – Enable or disable Brute Force Protection.
   - **Maximum Login Attempts** – Number of unsuccessful login attempts after which an account should be blocked.
   - **Cooldown Time** – Time in seconds after which a login using the locked account should be possible again after the account has been locked out of the system.
2. Click **Send Changes** and **Activate**.

Handling Forced Key Authentication

For various administrative purposes (such as collecting statistics with external tools), it may be desired to randomly connect to a system with an external SSH client, thereby omitting user interaction. Using the Microsoft Management Console (MMC), you can export a private key in encrypted PFX file format from the Certificate Store. However, this file is not usable by the Barracuda CloudGen Firewall. You must convert the PFX file to an unencrypted private key in PEM format.

Step 1. Create an Administrative Login

1. Go to **CONFIGURATION > Configuration Tree > Box > Administrators**.
2. Click **Lock**.
3. Click **+** to add a new administrative account.
4. Enter a name for the account and click **OK**.
5. In the **Administrator Authentication** section of the **Administrators** window, set **Authentication Level** to *Key*.
6. Import the public RSA key that has been issued for this user from the Microsoft Certificate Management Store.

Step 2. Export the Private Key from the Certificate Management Store

1. On the Windows client, open the Certificate Management Store. At the DOS prompt, enter:
   - `C:\windows\system32\certmgr.msc`
2. Browse to **Personal** > **Certificates**.
3. Right-click the certificate and select **All Tasks** > **Export**.
4. In the **Certificate Export Wizard**, select **Yes** to export the private key.
5. In the **PKCS #12** tab, clear the **Enable strong protection** check box.
6. Enter a password.
7. Specify a file name. For example: `private_key.pfx`

Step 3. Copy the PKCS12 (.pfx) File to a UNIX Client Supporting OpenSSL

Copy the PFX file to a UNIX client that supports OpenSSL, such as the Barracuda CloudGen Firewall.

Step 4. Convert the RSA Key from PKCS12 Format to PEM Format (encrypted)

On the UNIX client, browse to the RSA key. At the command line, enter:

- `# openssl pkcs12 -in private_key.pfx`
- `-nocerts -out priv.key`

where `priv.key` specifies the file name after conversion.

Step 5. Extract the Private Key and Generate an OpenSSH SSH-2 Private Key (unencrypted)

At the command line, enter:

- `# openssl rsa -in priv.key > ~/.ssh/`
- `id_rsa_my_priv_key`

where `id_rsa_my_priv_key` specifies the file name after decryption, and `~/.ssh/` is an arbitrarily chosen path on the UNIX client.

Step 6. Log into the Barracuda CloudGen Firewall

At the command line, enter:

- `# ssh -i ~/.ssh/id_rsa_my_priv_key`
- `-lloginname dest-ip`

where `loginname` specifies the name of the administrative account as defined in **Step 1**, and `dest-ip` specifies the Barracuda CloudGen Firewall's login IP address.

Depending on the client that the key was converted on, you may need to change the file permissions

of the private key file. If the gateway refuses to use the key, change the file permissions of the key by entering:

- `chmod 600 ~/.ssh/id_rsa_my_priv_key`

You can use the transformed private key with third-party remote SSH clients. For example, you can use it with SSH agents or import it into PuTTYGen for conversion into the file format for PuTTY (.ppk).