Opening Ports Through WHM and Listening to Data from a Third-Party Device

Introduction :

This guide provides comprehensive steps to open ports through WHM using the ConfigServer Security & Firewall (CSF) plugin and set up your server to listen for data transmitted by third-party devices.

Step 1: Installing ConfigServer Security & Firewall (CSF) on WHM

If the ConfigServer Security & Firewall plugin is not already available in your WHM, follow these steps to install it:

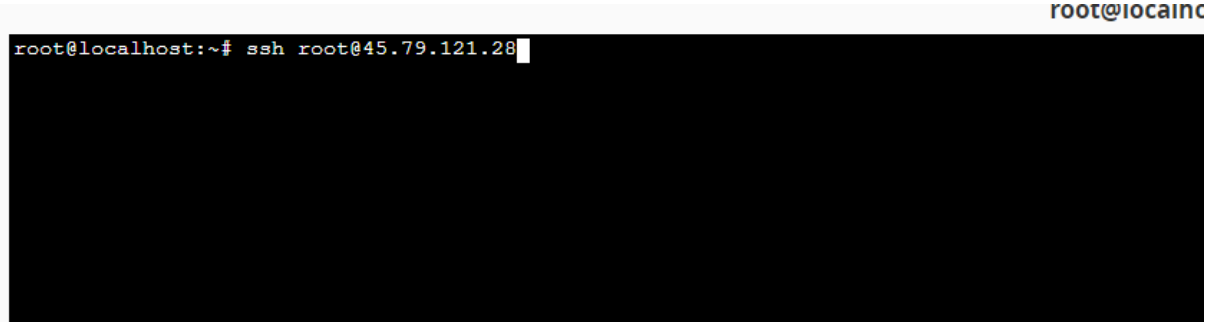
1. Access the Server via SSH

- Open an SSH terminal and connect to your server.

- Command: `ssh root@your-server-ip`

- Enter the root password when prompted.

Screenshot 1: SSH Terminal Connecting to Server



2. Navigate to the Source Directory

- Command: `cd /usr/local/src/`

Screenshot 2: Navigating to Source Directory

(Insert a screenshot showing the terminal where you're navigating to the source directory.)

3. Download the CSF Installation Package

- Command: `wget https://download.configserver.com/csf.tgz`

Screenshot 3: Downloading the CSF Package

(Insert a screenshot of the terminal output after running the wget command.)

4. Extract the Package

- Command: `tar -xzf csf.tgz`

Screenshot 4: Extracting CSF Package

(Insert a screenshot of the extraction process showing the contents of the package.)

5. Change to the CSF Directory

- Command: `cd csf`

6. Run the CSF Installation Script

- Command: `sh install.sh`

Screenshot 5: Running the Installation Script

(Insert a screenshot showing the installation process.)

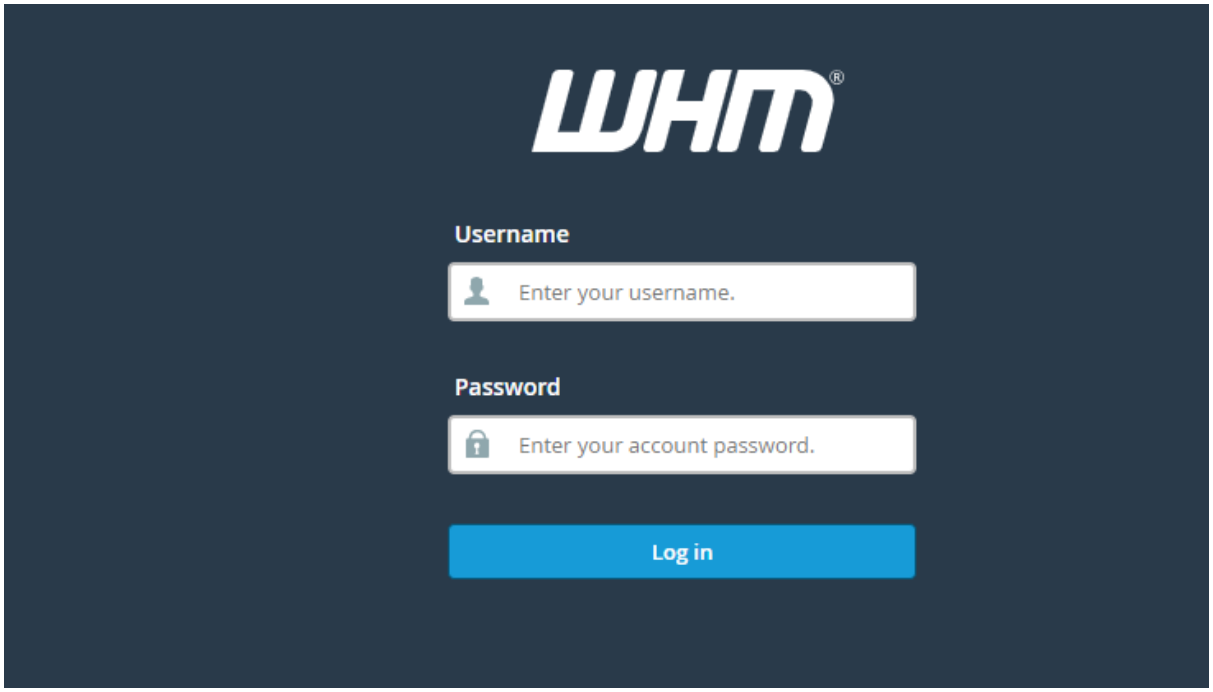
Step 2: Opening Ports on WHM Using CSF

Once CSF is installed, follow these steps to open the necessary ports:

1. Log into WHM

- Access WHM using the root user credentials.

Screenshot 6: WHM Login Screen

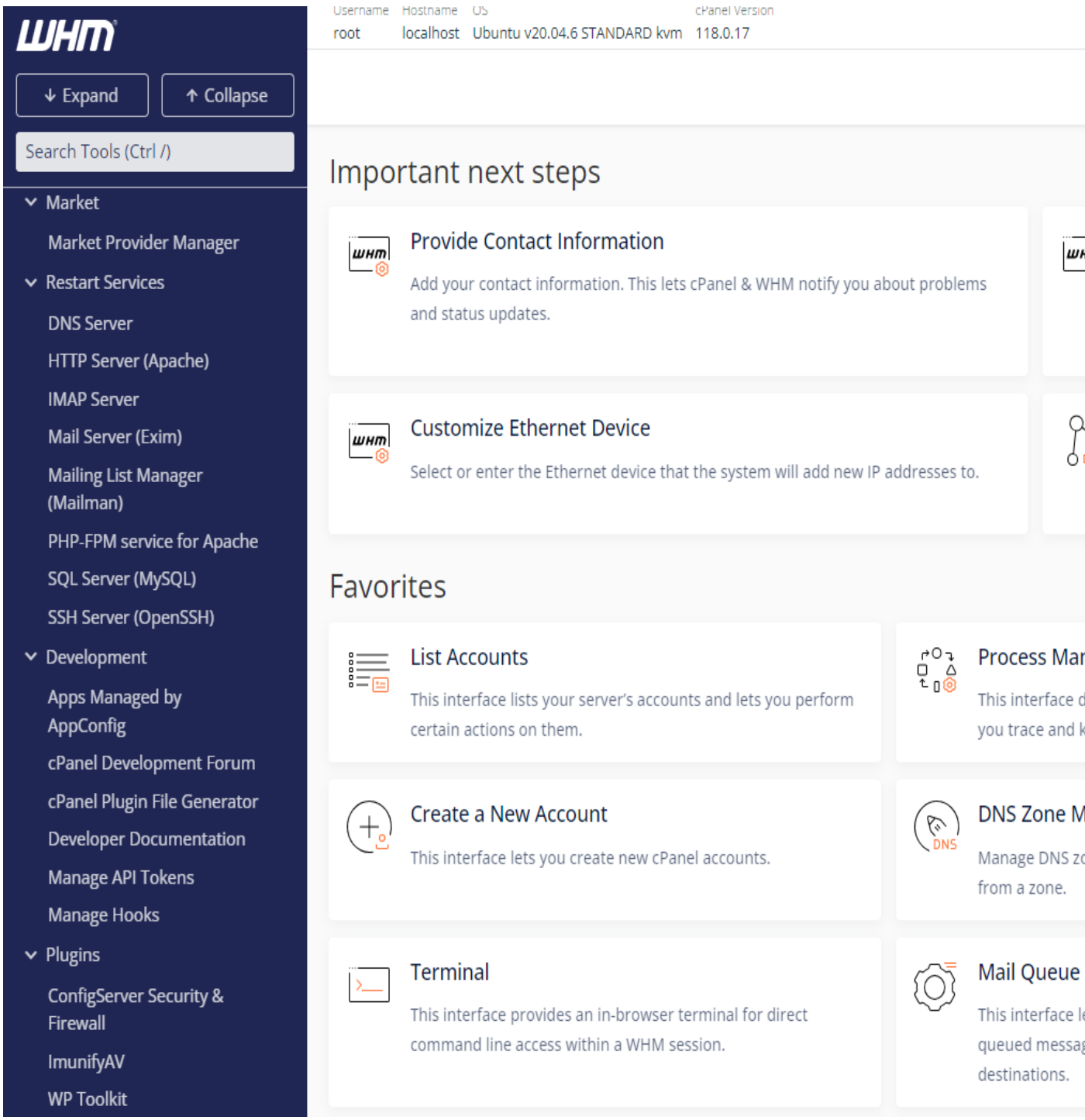


2. Navigate to the CSF Plugin

- Go to Home > Plugins > ConfigServer Security & Firewall.

Screenshot 7: CSF Plugin Navigation

Select Plugin and under plugin select ConfigServer Security & Firewall



3. Configure Firewall Settings

- In the "csf - ConfigServer Firewall" section, click on the Firewall Configuration button.

Screenshot 8: Firewall Configuration Button

4. Open Incoming TCP Port

- Scroll down to the "Allow incoming TCP ports" section.
- Enter the port number in the TCP_IN textbox.

Select Firewall Configuration

csf - ConfigServer Firewall	
Firewall Configuration	Edit the configuration file for the csf firewall and lfd
Firewall Profiles	Apply pre-configured csf.conf profiles and backup/restore csf.conf
View iptables Rules	Display the active iptables rules
Search for IP	Search iptables for IP address
Firewall Allow IPs	Edit csf.allow, the IP address allow file (Currently: 0 permanent IP allows)
Firewall Deny IPs	Edit csf.deny, the IP address deny file (Currently: 200 permanent IP bans)
Firewall Enable	Enables csf and lfd if previously Disabled
Firewall Disable	Completely disables csf and lfd
Firewall Restart	Restart the csf iptables firewall
Firewall Quick Restart	Have lfd restart the csf iptables firewall
Temporary Allow/Deny	Temporarily block IP address to port(s) * for seconds. Comment: (ports can be either * for all ports, a single port, or a comma separated list of ports)

5. Open Outgoing TCP Port (If Needed)

- Scroll down to the "Allow outgoing TCP ports" section.
- Enter the port number in the TCP_OUT textbox.

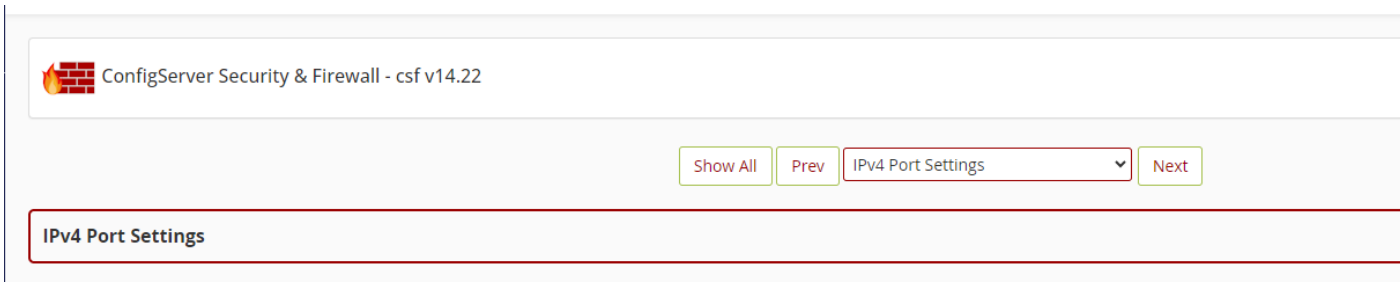
6. Open Incoming UDP Port (If Needed)

- Scroll down to the "Allow incoming UDP ports" section and enter the port number in the UDP_IN textbox.

7. Open Outgoing UDP Port (If Needed)

- Enter the port number in the UDP_OUT textbox for outgoing UDP data.

Select IPv4 Port Settings



Sroll down on same page and you will get this is very important look at this

If LF_SPI off the on it

Add Port number In TCP_IN and TCP_OUT for example you have to open port 7009

Then add both of them in that page

This will force incoming DNS traffic only through port 53

Disabling this option will break firewall functionality that relies on stateful packet inspection (e.g. DNAT, PACKET_FILTER) and makes the firewall less secure

This option should be set to "1" in all other circumstances

LF_SPI = Off On

Allow incoming TCP ports

TCP_IN = 20,21,22,25,53,80,110,143,443,465,587,853,993,995,2077,2078,2079,2080,2082,2083,2086,2087,2095,2096,8443,7001,9000,7009

Allow outgoing TCP ports

TCP_OUT = 20,21,22,25,37,43,53,80,110,113,443,587,853,873,993,995,2086,2087,2089,2703,7001,9000,7009

Allow incoming UDP ports

UDP_IN = 20,21,53,80,443,853

Allow outgoing UDP ports
To allow outgoing traceroute add 33434:33523 to this list

UDP_OUT = 20,21,53,113,123,853,873,6277,24441

8. Save and Apply Changes

- Scroll to the bottom of the page and click the Change button.

- Click the Firewall Restart button to apply the new settings.

To disable rate limiting set to "0", otherwise set according to the iptables documentation for the limit module. For example, "1/s" will limit to one packet per second

ICMP_OUT_RATE = 0

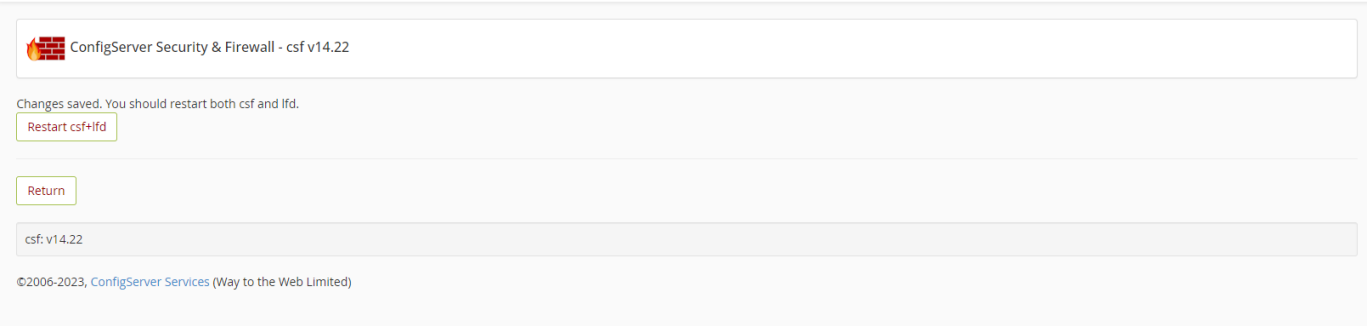
For those with PCI Compliance tools that state that ICMP timestamps (type 13) should be dropped, you can enable the following option. Otherwise, there appears to be little evidence that it has anything to do with a security risk and can impact network performance, so should be left disabled by everyone else

ICMP_TIMESTAMPDROP = 0

Show All | Prev | IPv4 Port Settings | Next

Change

Return



Step 3: Setting Up the Server to Listen for Data

After opening the necessary ports, set up your server to listen for incoming data from the third-party device:

1. Create the Socket Server Script

Write a PHP script to listen on the port and process incoming data. Below is an example:

```php
<?php
$host = '0.0.0.0'; // Listen on all IP addresses
$port = 12345; // Replace with the port you opened

// Create a TCP Stream socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0);
if (!$socket) {
    die('Could not create socket: ' . socket_strerror(socket_last_error()));
}

// Bind the socket to the port
```

```
if (!socket_bind($socket, $host, $port)) {
    die('Could not bind to port: ' . socket_strerror(socket_last_error()));
}

// Start listening for connections
socket_listen($socket, 5);

echo "Listening on $host:$port...\n";

// Accept incoming connections
while (true) {
    $client = socket_accept($socket);
    if ($client) {
        $input = socket_read($client, 1024);
        echo "Received data: $input\n";
        // Process the data as needed
        socket_close($client);
    }
}

// Close the main socket
socket_close($socket);
?>
```

Screenshot 12: PHP Script for Socket Server

(Insert a screenshot showing the PHP code in a code editor.)

2. Save the Script

- Save the script as `socket_server.php` in the `/home/username` directory on your server.

3. Run the Socket Server

- Use the following command in the terminal to start the socket server:

```bash
php /home/username/socket_server.php
```

Screenshot 13: Running the Socket Server

(Insert a screenshot showing the terminal output when the socket server is running.)

4. Set Up a Cron Job for Continuous Operation

- To ensure the socket server runs continuously, set up a cron job:

```bash
/usr/bin/php /home/username/socket_server.php >/dev/null 2>&1
```

Screenshot 14: Cron Job Setup

(Insert a screenshot showing the cron job setup in WHM.)

Conclusion

By following these steps, you will have successfully opened the necessary ports on your WHM server and set it up to listen for and process data from a third-party device. This setup is essential for applications requiring real-time data transmission and processing.

Next Steps:

- Add relevant screenshots for each step following the guide above.
- Ensure the screenshots are clear and correspond to the instructions provided.

Let me know if you need further customization or additional details!