cPHulk Brute Force Protection

(*WHM >> Home >> Security Center >> cPHulk Brute Force Protection*)

Overview
Enable cPHulk
Configure cPHulk
Example behavior
Additional documentation

Overview

This interface allows you to configure cPHulk, a service that provides protection for your server against brute force attacks. A brute force attack uses an automated system to guess the password of your web server or services.

cPHulk monitors the following web servers and services:

- cPanel services (Port `2083`).
- WHM services (Port `2087`).
- Mail services (Dovecot® and Exim).
- The PureFTPd service.
- Secure Shell (SSH) access.

When cPHulk blocks an IP address or account, it does **not** identify itself as the source of the block. Instead, the login page displays the following warning message: *The login is invalid.*

Important:

We **strongly** recommend that you add your own IP address or addresses to the whitelist to avoid a lockout of the `root` user account.

Notes:

- cPHulk does **not** affect public key authentication to the server. If cPHulk locks an account or all accounts out of the server, you may still use public keys, API tokens, and access hashes to authenticate to your server.
- cPHulk does **not** consider multiple login attempts that use the same IP address, username, **and** password as separate failures if they occur within the same six-hour period.
- To manage cPHulk from the command line, read our cPHulk Management on the Command Line documentation.
- The *Create Support Ticket* interface (*WHM >> Home >> Support >> Create Support Ticket*) automatically adds cPanel Support's IP addresses to cPHulk's whitelist.

Enable cPHulk

To enable cPHulk on the server, set the toggle to *On*.

Notes:

- By default, your server sets the `UseDNS` setting to `enabled` in the `/etc/ssh/sshd_config` file. The `UseDNS` setting sends the hostname to the Password Authentication Module (PAM), which ships with cPanel & WHM, for SSH session authentication. cPHulk also requests authentication information from the PAM to determine whether a login attempt could be a brute force attack.
- If an attacker spoofs a DNS pointer record to impersonate a trusted hostname, the `UseDNS` setting and cPHulk's whitelist will conflict. This allows the attacker to perform a brute force attack against the server with unlimited login attempts. Therefore, the system disables the `UseDNS` setting when you enable cPHulk.

Configure cPHulk

Click a tab below for more information about those cPHulk settings:

You can configure the following *Configuration Settings* options:

Username-based Protection

Setting	Description	Default
Username-based Protection	Whether to enable the username-based protection settings. Set the toggle to *On* to enable the *Username-based Protection* setting. Username-based protection tracks login attempts for user accounts. When you disable cPHulk, existing account locks will remain. ⚠ **Note:** You **must** click *Save* to implement any change to this setting.	*On*
Brute Force Protection Period (in minutes)	The number of minutes during which cPHulk measures all login attempts to a specific user's account. • If several attackers attempt to log in, and they reach the account's *Maximum Failures by Account* value within this period, cPHulk classifies this as a brute force attempt. • cPHulk blocks logins from **any** IP addresses to that account, regardless of the attackers' IP address or addresses. • Enter a value between `1` and `1,440` for this setting.	5
Maximum Failures by Account	The maximum number of failures that cPHulk allows per account within the *Brute Force Protection Period (in minutes)* time range. • If a brute force attack meets this number of attempts, the system locks the account, regardless of the attackers' IP addresses. • cPHulk locks the account for one minute for each attempt that you allow with this setting. For example, if you set the *Maximum Failures by Account* setting to `15`, after 15 login attempts cPHulk locks the account for 15 minutes. • When you set this value to `0`, cPHulk blocks **all** login attempts (this includes the `root` account). To avoid this lock-out, you **must** whitelist your IP address.	`15`
Apply protection...	Select one of the following options to control how cPHulk applies its protection: • *Apply protection to local addresses only* — Limit username-based protection to trigger **only** on requests that originate from the local system. This ensures that a user cannot brute force other accounts on the same server. • *Apply protection to local and remote addresses* — Allow username-based protection to trigger for all requests, regardless of their origin.	This setting defaults to *Apply protection to local addresses only*.
Allow username protection to lock the "root" user	Whether to apply username-based protection rules to the `root` user.	This checkbox defaults to deselected.

IP Address-based Protection

Setting	Description	Default
IP Address-based Protection	Whether to enable the IP address-related protection settings. Set the toggle to *On* to enable the *IP Address-based Protection* setting. IP address-based protection tracks login attempts from specific IP addresses. When you disable cPHulk existing account locks will remain. ⚠ **Note:** You **must** click *Save* to implement any change to this setting.	*On*

IP Address-based Brute Force Protection Period (in minutes)	The number of minutes during which cPHulk measures all login attempts from an attacker's IP address. cPHulk classifies the following as a brute force attack: • Attackers on a specific IP address attempt to log in repeatedly with different usernames or passwords. • They reach the *Maximum Failures per IP Address* value. ⚠ **Notes:** • cPHulk measures the attacker's IP address for the number of minutes that you specify. • cPHulk will **not** measure **all** IP addresses.	`15`
Maximum Failures per IP Address	The maximum number of times that a potential attacker at a specific IP address can fail to log in before cPHulk blocks that IP address. When you set this value to `0`, cPHulk blocks **all** login attempts (this includes the `root` account). To avoid this lock-out, you **must** whitelist your IP address.	`5`
Command to Run When an IP Address Triggers Brute Force Protection	The full path to a command that you want the system to run when an IP address triggers brute force protection. For a list of variables to use in this command, read the Command variables section below.	(none)
Block IP addresses at the firewall level if they trigger brute force protection	Whether you wish to automatically add IP addresses that trigger brute force protection to the firewall. ⚠ **Notes:** • This option writes a new `iptables` rule and requires `iptables` version 1.4 or higher to block IP addresses at the IP address-based level. • This option does **not** exist on Virtuozzo.	This checkbox defaults to deselected.

One-day Blocks

Setting	Description	Default
Maximum Failures per IP Address before the IP Address is Blocked for One Day	The maximum number of times that a potential attacker at a specific IP address can fail to log in before cPHulk blocks that IP address for a one-day period.	`30`
Command to Run When an IP Address Triggers a One-Day Block	The full path to a command that you want the system to run when the system blocks an IP address for a one-day period. For a list of variables to use in this command, read the Command variables section below.	(none)
Block IP addresses at the firewall level if they trigger a one-day block	Whether you wish to automatically add IP addresses that trigger a one-day block to the firewall. This option writes a new `iptables` rule and requires `iptables` version 1.4 or higher. ⚠ **Notes:** • This option writes a new `iptables` rule and requires `iptables` version 1.4 or higher to block IP addresses at the IP address-based level. • This option does **not** exist on Virtuozzo.	This checkbox defaults to selected.

Login History

Setting	Description	Default
Duration for Retaining Failed Logins (in minutes)	The number of minutes that the system allows for an attacker to reach the following settings: • *Maximum Failures by Account* • *Maximum Failures per IP Address* • *Maximum Failures per IP Address before the IP Address is Blocked for One Day* This setting also determines for how long the system displays failed login entries on the *History Reports* tab.	`360`

Notifications

Setting	Description	Default
Send a notification upon successful root login when the IP address is not on the whitelist	Whether you wish to receive a notification when the `root` user successfully logs in from an IP address that does not exist in the whitelist. **Note:** The system only sends a notification once in any 24-hour window for a specific username, service, and IP address combination.	This checkbox defaults to deselected.
Send a notification upon successful root login when the IP address is not on the whitelist, but from a known netblock	Whether you wish to receive a notification when the `root` user successfully logs in from an IP address that does not exist in the whitelist, but exists in a known netblock.	This checkbox defaults to deselected.
Send a notification when the system detects a brute force user	Whether you wish to receive a notification when cPHulk detects a brute force attack.	This checkbox defaults to selected.

Note:

Click *Save* to save your settings.

Command variables

You can use the following variables in commands that you enter for the *Command to Run When an IP Address Triggers Brute Force Protection* and *Command to Run When an IP Address Triggers a One-Day Block* settings:

Variable	Description
`%exptime%`	When cPHulk will release the ban.
`%max_allowed_failures%`	The maximum number of allowed failures to trigger cPHulk (excessive or non-excessive failures).
`%current_failures%`	The number of current failures.
`%excessive_failures%`	When the one-day block triggers, this boolean becomes true.
`%reason%`	The reason for the ban.
`%remote_ip%`	The IP address to ban.
`%authservice%`	The last service to request authentication (for example, `webmaild`).
`%user%`	The last username to request authentication.
`%logintime%`	The time of the request.
`%ip_version%`	The IP version, either IPv4 or IPv6.

Example behavior

The following table contains variables for different hacking scenarios, and cPHulk's response if you use the default settings:

Scenario					cPHulk's response
Address	**Account**	**Password**	**Attempts**	**Time range**	
`192.168.0.1`	`username`	N/A	One.	N/A	No response.
`192.168.0.1`	`username`	The same password each time.	Five or more.	365 minutes.	No response.
`192.168.0.1`	`username`	Different passwords each time.	Five to nine.	Five minutes.	Lock the `username` account for five minutes.
`192.168.0.1`	`username`	Different passwords each time.	Five or more.	365 minutes.	No response.
`192.168.0.1`	`username`	Different passwords each time.	10 to 29.	Five minutes.	Block `192.168.0.1` for 15 minutes.
`192.168.0.1`	`username`	Different passwords each time.	30 or more.	Five minutes.	Block `192.168.0.1` for two weeks.
Various	`username`	N/A	Five or more.	Five minutes.	Lock the `username` account for five minutes.
Various	Various	N/A	Five or more.	Five minutes.	No response.
`192.168.0.1`	Various	N/A	Five to nine.	Five minutes.	No response.
`192.168.0.1`	Various	N/A	10 to 29.	Five minutes.	Block `192.168.0.1` for 15 minutes.
`192.168.0.1`	Various	N/A	30 or more.	Five minutes.	Block `192.168.0.1` for two weeks.

Note:

The settings that you choose determine cPHulk's behavior in these scenarios.

Additional documentation

- cPHulk Brute Force Protection
- Compiler Access
- Security Advisor
- Tweak Settings - Security
- Host Access Control