

FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

The Firewall That Was Never On — CSF Test Mode and 150 Full Apache Workers

A field story about a cPanel server whose firewall looked configured but had never been running — the one line almost nobody checks — and the five-minute check I now run before trusting any CSF setting.

Daanial Farrukh · 21 October 2026 · <https://quabin.com/csf-test-mode-firewall-never-on-apache-workers/>

The report was simple. The site had been stuck for about ten minutes, then came back on its own.

A customer's cPanel server. Ubuntu 24.04, one busy public website, CSF installed like on every cPanel box. The slowdown had lasted about ten minutes, late in the morning, and then everything was normal again.

The first look at the Apache error log made it worse, not better. At 11:17 — right in the middle of the bad window — there was nothing. Not one line.

"Empty log. So the server was fine at 11:17?"

No. It was not fine. It just had nothing new to say.

I scrolled up instead of grepping one minute. Seven minutes earlier, at 11:09:59, there it was:

```
AH00161: server reached MaxRequestWorkers setting, consider raising the MaxRequestWorkers setting
```

Apache prints that line **once per event**, not once per minute. The server hit the wall at 11:09:59 and stayed against it, silently, until someone restarted Apache at 11:20:07.

So the question became: what pushed 150 workers to the wall in under five seconds?

And a second question, which turned out to matter more: why didn't the firewall stop it?

The Trap — A Firewall You Can See but Isn't There

That's the trap. CSF in Test Mode looks exactly like a working firewall. It shows in WHM, every setting is editable, the rules even load. **It just doesn't stay on, and its brain never starts.**

A firewall you have never tested is a firewall you do not have.

What Happened in Those Two Seconds

The domain's own access log told the story. On cPanel, HTTPS traffic for a site goes to its own file, and the plain one may be empty — that cost me a few minutes too:

```
ls -la /usr/local/apache/domlogs/ | grep example.com
```

The file with the data was `example.com-ssl_log`. Count the top talkers in the minute before the alarm:

```
grep '28/Aug/2026:11:09' /usr/local/apache/domlogs/example.com-ssl_log | awk '{print $1}' |  
sort | uniq -c | sort -rn | head -5
```

One IP stood out. A virtual machine in a public cloud, nothing to do with the site's audience. Between 11:09:55 and 11:09:56 it sent **542 requests in about two seconds**.

What it asked for tells you exactly what it was:

- `wp-config.php.bak`
- `.vscode/sftp.json`
- `WEB-INF/web.xml`
- `values.yaml`

A config-file scanner. It was looking for forgotten backups and developer files that hold passwords. Every single request returned 404. Nothing leaked.

But a 404 is not free. Requests ending in `.php` were handed to PHP-FPM before anyone knew the file didn't exist, and the error log filled with this:

```
AH01071: Got error 'Primary script unknown'
```

Each of those requests held an Apache worker while it waited. 542 requests arriving at once, against a pool of 150 workers. Three seconds later: AH00161.

Why the Server Didn't Look Down

Here's the part that trips people up. Apache was not dead. Between 11:10 and 11:19 it still served **1,218 requests**. The top real visitor in that window made only 84 hits.

So from the outside, some pages loaded and many hung. From the inside, Apache was alive, the error log was quiet, and the process list looked normal. Every quick health check said "up".

Absence of errors is not health.

We also ruled one thing out before moving on. Someone suspected IPv6 traffic slipping past the rules. Every client IP in the logs was IPv4, so that theory died in one `grep`.

What CSF Test Mode Actually Is

CSF Test Mode is a safety switch that clears the firewall every five minutes and never starts the login-failure daemon.

It exists for a good reason. When you install CSF on a remote server, one wrong port list can lock you out of SSH. So CSF ships with `TESTING = "1"`. In that state:

- A cron job flushes iptables every 5 minutes. Whatever rules loaded are gone again.
- `lfd`, the Login Failure Daemon — the part that blocks brute force, port scans and connection floods — does not run at all.
- WHM still shows the full ConfigServer screen, so nothing looks wrong.

The idea is that you test, confirm you can still log in, and then switch it off. On this server, nobody ever switched it off.

The flow everyone assumed:

Scanner → CSF connection limit → blocked after 30 connections → Apache stays free

The flow that actually ran:

Scanner → no firewall rules → Apache → PHP-FPM → 150 workers full → real visitors wait

Where Test Mode is common:

- Servers where CSF was installed by a script during provisioning and never revisited.
- Boxes migrated from another host, where the old firewall was trusted and the new one was "set up later".
- Servers managed by more than one team, where each assumes the other finished the job.

Why Nobody Sees It in Testing

Because the settings look right. You open WHM › Plugins › ConfigServer Security & Firewall, you see your ports, your allow list, your rules. Nobody reads the very first line of the configuration — and the warning banner at the top of the CSF page is easy to scroll past once you have seen it a few times.

And because a quiet server never tests the firewall. Until the day a scanner arrives, a firewall that does nothing and a firewall that works look exactly the same.

How to Actually Fix This

Step 1: Check Whether CSF Is in Test Mode

Read-only:

```
grep '^TESTING ' /etc/csf/csf.conf
```

TESTING = "1" means Test Mode. Your firewall is being wiped every five minutes. TESTING = "0" is what you want.

Then check the daemon:

```
systemctl is-active lfd
```

inactive or failed on a server you believed was protected is the same finding from another angle.

Step 2: Prove You Will Not Lock Yourself Out

This is the step people skip, and it is why Test Mode exists. Before you switch it off, confirm the ports you need are open:

```
grep '^TCP_IN ' /etc/csf/csf.conf
```

On this server I confirmed 22, 80, 443 and 2087 were all in the list before touching anything. If your SSH runs on a custom port, that port must be there. Also add your own admin IP to /etc/csf/csf.allow so a

mistake costs you nothing.

Step 3: Confirm the Kernel Supports Connection Limits

CONNLIMIT needs the `xt_connlimit` module. CSF ships a tester — in WHM it is the **Test iptables** button on the CSF page:

```
/usr/local/csf/bin/csfctest.pl
```

Look for Testing `xt_connlimit`...OK and the final line `RESULT: csf should function on this server`. If `connlimit` fails, the setting below will do nothing.

Step 4: Set a Per-IP Connection Limit

This is a change. Back up first:

```
cp -p /etc/csf/csf.conf /etc/csf/csf.conf.bak-$(date +%F)
```

In WHM > Plugins > ConfigServer Security & Firewall > Firewall Configuration, set:

```
CONNLIMIT = "80;30,443;30"
```

It was blank on this server. The format is `port;limit`, comma-separated. This allows each single IP 30 concurrent connections to port 80 and 30 to port 443. A real browser opens a handful. A scanner firing 542 requests in two seconds does not get past 30.

Step 5: Switch Test Mode Off

Same screen: set **TESTING** to **Off**, then **Change** and **Restart csf+lf**. From the shell, the same change:

```
sed -i 's/^TESTING = "1"/TESTING = "0"/' /etc/csf/csf.conf
```

```
csf -ra
```

Step 6: Verify

This step is where most tutorials stop and where mine starts.

```
grep '^TESTING ' /etc/csf/csf.conf
```

It must say "0".

```
systemctl is-active lfd
```

It must say active.

```
iptables -S | grep connlimit
```

You want to see rules with `--connlimit-above 30` on ports 80 and 443. If that command returns nothing, the limit is not in the live firewall — whatever the settings page says.

Then wait more than five minutes and run the last command again. In Test Mode the rules disappear after the cron runs. If they are still there, the firewall is finally staying on.

The First Block Was a Real User

Expect this. The firewall had never run, so it had never blocked anyone. A few hours after Test Mode was switched off, lfd made its **first block ever** on this server.

It was not an attacker. It was a staff PC in the customer's own office, banned permanently by LF_SMTPAUTH — five failed SMTP logins in an hour. A mail client had a wrong saved password and kept retrying it in the background, as it probably had for months. Before, nothing noticed. Now the firewall did.

What gave it away: the same IP was the site's top real visitor that morning, with those 84 hits. Attackers don't read the company website all morning.

To prove it, one command shows the mailbox and the client name behind the failures:

```
grep '<BLOCKED_IP>' /var/log/exim_mainlog | grep -i "authenticator failed"
```

Unblock with **Quick Unblock** on the CSF page, or:

```
csf -dr <BLOCKED_IP>
```

Then fix the saved password in the mail client. Unblocking alone only buys you an hour — the client retries, fails five times, and the ban comes back.

This error is not a bug. It is the verification. The firewall is finally doing its job.

What I Have Not Proven Yet

I want to be honest about one gap. Workers stayed stuck for the full ten minutes, long after the scanner had finished in two seconds. My theory is a small PHP-FPM pool — too few PHP processes, so a backlog of PHP requests kept Apache workers waiting. pm.max_children was never checked on this server, so I am labelling it a theory, not a finding. I have seen that exact limit take down another site; I wrote about it in *The Outage That Wasn't a DDoS*.

To confirm it, check the pool for the busiest domain:

```
grep -r 'max_children' /opt/cpanel/ea-php*/root/etc/php-fpm.d/
```

And read the PHP-FPM error log for the line server reached pm.max_children setting at the same minute.

The Bigger Lesson — A Five-Minute Firewall Check

On every cPanel server I now inherit, before I believe any firewall setting:

- 1 **Is CSF in Test Mode?** `grep '^TESTING ' /etc/csf/csf.conf`
- 2 **Is lfd running?** `systemctl is-active lfd`
- 3 **Are the rules actually loaded?** `iptables -S | wc -l` — a handful of lines means no firewall.
- 4 **Is there a per-IP limit?** `grep '^CONNLIMIT ' /etc/csf/csf.conf`
- 5 **Does the kernel support it?** `/usr/local/csf/bin/csftest.pl`
- 6 **Where does Apache really log?** `ls /usr/local/apache/domlogs/` — read the `-ssl_log` file.
- 7 **When did the workers run out?** `grep AH00161 /usr/local/apache/logs/error_log | tail -3` — then read around the timestamp, not the minute someone complained.

Plain takeaway: the settings page tells you what someone intended. Only iptables tells you what is running.

What I Tell Every Customer Now

When a customer says "we have a firewall", I don't ask which one. I ask:

"When did your firewall last block something?"

If the answer is a date from last week, they're in good shape. If the answer is "I don't know — it's installed", we're not done yet.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

What is CSF Test Mode?

It is the default state of ConfigServer Security & Firewall after install, set by `TESTING = "1"` in `/etc/csf/csf.conf`. In Test Mode a cron job clears the iptables rules every five minutes and `lfd`, the login failure daemon, does not run. It is meant to prevent lockouts during setup, not to stay on.

How do I check if CSF is actually protecting my server?

Run `grep '^TESTING ' /etc/csf/csf.conf` and expect `"0"`, then `systemctl is-active lfd` and expect `active`. Finally run `iptables -S` and confirm the rules are present, and check again after five minutes. If the rules vanish, Test Mode is still on.

What does CONNLIMIT do in CSF?

`CONNLIMIT` caps how many concurrent connections a single IP can hold to a port. `CONNLIMIT = "80;30,443;30"` allows 30 connections per IP to each web port, which is plenty for real browsers but stops a scanner from taking every Apache worker. It needs the `xt_connlimit` kernel module, which CSF's Test iptables check confirms.

Why is my Apache error log empty while the site is down?

Apache writes `AH00161: server reached MaxRequestWorkers once` when the limit is hit, not every minute while it stays full. A quiet log during an outage often means the problem started earlier. Read backwards from the time of the complaint instead of grepping a single minute.

Why did CSF block a real user right after I disabled Test Mode?

Because `lfd` is running for the first time and sees failures it ignored before. The most common case is a mail client with a stale saved password that keeps failing SMTP authentication and trips `LF_SMTPAUTH`. Check the Exim log for "authenticator failed" from that IP, fix the password in the client, then unblock.

Read online: <https://quabin.com/csf-test-mode-firewall-never-on-apache-workers/>