ISPProtect BanDaemon 2.1.1 - Complete User Guide

Table of Contents

1. Overview
2. Installation
3. License Management
4. Configuration
5. Command Line Usage
6. Service Management
7. Monitoring & Reporting
8. IP Management
9. Advanced Features
10. Troubleshooting
11. Best Practices

Overview

ISPProtect BanDaemon is a powerful Linux security daemon that automatically protects your server from brute force attacks, unauthorized access attempts, and malicious activities. It monitors log files in real-time and automatically bans offending IP addresses using iptables firewall rules.

Key Features

- **Real-time Protection**: Monitors log files continuously for suspicious activities
- **Automatic IP Banning**: Uses iptables to block malicious IPs instantly
- **Multi-Service Support**: Protects SSH, web servers, email services, DNS, and more
- **IPv6 Ready**: Full support for both IPv4 and IPv6 networks
- **Distributed Operation**: Share ban information across multiple servers
- **Evidence Collection**: Automatically preserves attack evidence for analysis
- **GeoIP Integration**: Country-based risk scoring and filtering
- **Abuse Reporting**: Automatic notification to ISP abuse contacts
- **Whitelist Protection**: Prevent accidental blocking of trusted IPs

System Requirements

- **Operating System**: Linux with iptables support
- **PHP**: Version 5.6 or higher (supports up to PHP 8.x)
- **Database**: MySQL 5.5+ or MariaDB 10.0+
- **Memory**: Minimum 64MB RAM (256MB recommended)
- **Disk Space**: 100MB free space minimum
- **Network**: Root access for iptables management

Installation

Quick Installation (Recommended)

For modern Linux distributions with systemd:

```
# Extract the package
tar -xzf ispp_bandaemon.tar.gz
cd bandaemon_install

# Run the installer as root
sudo bash install.sh
```

Legacy Installation

For older systems using init.d:

```
# Extract the package
tar -xzf ispp_bandaemon.tar.gz
cd bandaemon_install

# Run the legacy installer as root
sudo bash install_initd.sh
```

Installation Process

The installer will:

1. **Check Requirements**: Verify PHP, MySQL, and system compatibility
2. **Create Database**: Set up the `ispprotect_bandaemon` database and user
3. **Install Files**: Copy program files to `/opt/ispprotect_bandaemon/`
4. **Configure Service**: Set up systemd service or init.d script
5. **License Setup**: Interactive prompt for license key or trial activation (new installations only)
6. **Create Symlink**: Make `ispp_bandaemon` command available system-wide
7. **Set Permissions**: Apply secure file permissions

Post-Installation Verification

```
# Check service status
systemctl status ispprotect_bandaemon

# Verify command availability
ispp_bandaemon --help

# Check log output
journalctl -u ispprotect_bandaemon -f
```

License Management

Interactive License Setup (New Installations)

Starting with version 2.0.0, ISPProtect BanDaemon includes an interactive license setup during fresh installations. The installer will prompt you with the following options:

```
License setup...

ISPProtect BanDaemon License Setup

Please choose one of the following options:

1. Enter your BanDaemon or ISPProtect 12 month scanner license key
2. Start a 14-day free trial (press Enter)

Enter your license key (or press Enter for trial):
```

For New Users (Trial License)

1. **Press Enter** to start a 14-day free trial
2. The installer creates an empty `license.key` file
3. All features are available during the trial period
4. After 14 days, purchase a permanent license at https://ispprotect.com

For Licensed Users

1. **Enter your license key** when prompted
2. The license key is automatically saved to the `license.key` file
3. Full access to all licensed features is immediately activated

Manual License Management

Installing a License Key Manually

If you need to install or update a license key after installation:

```
# Method 1: Direct file editing
sudo nano /opt/ispprotect_bandaemon/license.key

# Method 2: Echo command (replace YOUR_LICENSE_KEY with actual key)
echo "YOUR_LICENSE_KEY" | sudo tee /opt/ispprotect_bandaemon/license.key

# Method 3: Using a text editor
sudo vim /opt/ispprotect_bandaemon/license.key
```

Important: The license key should be entered exactly as provided, typically as a single line of text.

Setting Up a Trial License Manually

For legacy installations or manual setup:

```
# Create an empty license file for trial usage
sudo touch /opt/ispprotect_bandaemon/license.key

# Set proper permissions
sudo chmod 640 /opt/ispprotect_bandaemon/license.key
sudo chown root:root /opt/ispprotect_bandaemon/license.key
```

Verifying License Installation

After installing or updating a license:

```
# Check that the license file exists
ls -la /opt/ispprotect_bandaemon/license.key

# Verify file permissions (should be 640 root:root)
stat /opt/ispprotect_bandaemon/license.key

# Restart the service to apply license changes
sudo systemctl restart ispprotect_bandaemon

# Check service status for any license-related errors
sudo systemctl status ispprotect_bandaemon
```

License Types

BanDaemon License

- Core IP banning and firewall management
- Real-time log monitoring and analysis
- Basic reporting and statistics
- Essential security protection features

ISPProtect 12 Month Scanner License

- Complete vulnerability scanning capabilities
- Advanced threat detection algorithms
- Extended GeoIP and country-based filtering
- Premium abuse reporting features
- Priority technical support

Trial License (14 Days)

- Full access to all features for evaluation
- No feature limitations during trial period
- Automatic conversion to licensed version when key is added
- Ideal for testing and proof-of-concept deployments

License File Security

The license file contains sensitive licensing information and should be protected:

```
# Verify secure permissions
ls -la /opt/ispprotect_bandaemon/license.key
# Should show: -rw-r----- 1 root root

# File location (do not move)
/opt/ispprotect_bandaemon/license.key

# Backup license (recommended)
sudo cp /opt/ispprotect_bandaemon/license.key /root/license.key.backup
```

Troubleshooting License Issues

License Not Recognized

```
# Check file exists and has content
sudo cat /opt/ispprotect_bandaemon/license.key

# Verify no extra characters or whitespace
sudo hexdump -C /opt/ispprotect_bandaemon/license.key | head

# Check daemon logs for license errors
sudo journalctl -u ispprotect_bandaemon | grep -i license
```

Trial License Expired

```
# Remove trial license and install commercial license
sudo nano /opt/ispprotect_bandaemon/license.key
# Add your commercial license key

# Restart service
sudo systemctl restart ispprotect_bandaemon
```

Permission Issues

```
# Fix license file permissions
sudo chown root:root /opt/ispprotect_bandaemon/license.key
sudo chmod 640 /opt/ispprotect_bandaemon/license.key
```

```
# Restart service
sudo systemctl restart ispprotect_bandaemon
```

Purchasing Licenses

- **Official Website**: https://ispprotect.com
- **License Types**: Choose between BanDaemon or full ISPProtect scanner licenses
- **Support**: Technical support included with all licensed versions
- **Upgrades**: Easy upgrade path from BanDaemon to full ISPProtect licensing

---

Configuration

Main Configuration File

Edit `/opt/ispprotect_bandaemon/config.inc.php` to customize your installation:

```
<?php
// Database Configuration
define('DB_HOST', 'localhost');
define('DB_NAME', 'ispprotect_bandaemon');
define('DB_USER', 'ispp_bandaemon');
define('DB_PASS', 'your_secure_password');

// Firewall Configuration
define('BANDAEMON_TABLE', 'PREROUTING');  // or 'INPUT'
define('BANDAEMON_MODE', 'INSERT');       // or 'APPEND'
define('IPTABLES_LEGACY', false);         // Set to true to use iptables-legacy commands

// Logging Configuration
define('BANDAEMON_SYSLOG', false);        // Use syslog or file logging
define('BANDAEMON_VERBOSE', false);       // Enable detailed logging
define('BANDAEMON_LOGNAME', false);       // Log source file names

// Web Protection
define('BANDAEMON_WWW_ROOT', '/var/www'); // Web server document root

// Service Control
define('BANDAEMON_DISABLE_WATCH', '');    // Disable named watches (comma-separated)
define('BANDAEMON_ENABLE_WATCH', '');     // Enable ONLY these watches (comma-separated)
define('BANDAEMON_TESTMODE', false);      // Detect and log, but never touch iptables

// Scoring
define('BANDAEMON_COUNTRY_SCORES', '');   // e.g. 'CN:30,RU:30,DE:-10' (see GeoIP
        Integration)

// Evidence collection and abuse reporting
define('BANDAEMON_EVIDENCE', false);      // Write evidence files for banned IPs
define('BANDAEMON_EVIDENCE_SCORE', 100);  // Minimum score before evidence is collected
define('BANDAEMON_ABUSE_ADDRESS', false); // Look up and store the abuse contact

// Server identification (unique per server, required when sharing one database)
define('SERVER_IDENT', 'web-server-01');

// Whitelisting
define('WHITELISTED_IPS', '');            // Comma-separated IPs/subnets never banned
define('LOG_FLOOD_ATTEMPTS', true);       // Log hits from whitelisted IPs

// Email Configuration (Optional)
define('BANDAEMON_DAILY_REPORT', false);  // Set to an address to receive daily reports
define('EMAIL_SENDER', '');
define('EMAIL_SENDER_NAME', 'ISPProtect Security Team');

// SMTP Configuration (optional - local sendmail is used when unset)
//define('EMAIL_SMTP_HOST', 'smtp.example.com');
```

```
//define('EMAIL_SMTP_PORT', 587);
//define('EMAIL_SMTP_AUTH', true);
//define('EMAIL_SMTP_USER', 'your_email@example.com');
//define('EMAIL_SMTP_PASS', 'your_email_password');
//define('EMAIL_SMTP_ENCRYPTION', 'tls');  // false, 'tls' or 'ssl'
?>
```

> **Note:** Only the settings listed above are evaluated. `config.inc.php` is plain PHP, so an unknown `define()` causes no error - it is simply ignored. Check the spelling against this list if a setting appears to have no effect.

Service-Specific Configuration

BanDaemon ships with built-in rulesets (“watches”) for the usual ISPConfig services. You can add your own rulesets, or override a built-in one, by placing `.conf` files in `/opt/ispprotect_bandaemon/conf.d/`.

> **Important:** These files use a plain `key = value` syntax. They are *not* INI files and they are *not* fail2ban jails. Section headers such as `[myapp]` and fail2ban settings such as `maxretry`, `findtime`, `bantime` or `filter` are **not supported**; they are rejected with an error at startup. A ruleset is named with an `ident = ...` line.

Example: Custom Application Protection

Create `conf.d/100-myapp.conf`:

```
# unique name of this ruleset
ident = myapp

# log file to watch - one "log =" line per file, wildcards * and ** allowed
log = /var/log/myapp/access.log

# PHP regex (including delimiters), must capture a date and a host
regex = /^<REGEX:DATE>\s+\S+\s+myapp:\s+Failed login attempt from <REGEX:HOST>$/

# which firewall chains the banned IP is blocked in
scope = global

# ignore this many hits per minute before scoring starts (0-20)
reduce = 0

# multiplies the calculated score (0.1-10)
bantime_factor = 1
```

Then restart the daemon and confirm the ruleset was picked up:

```
systemctl restart ispprotect_bandaemon
journalctl -u ispprotect_bandaemon -n 50 | grep -i "config file"

# Expected:
# Read config file /opt/ispprotect_bandaemon/conf.d/100-myapp.conf and added myapp
        ruleset.
```

If instead you see `invalid setting ...` or `Missing ident line ...`, the ruleset was **not** loaded correctly and the log file is not being monitored.

Available Configuration Options

These are the **only** settings accepted in a `conf.d` file. Every other key is reported as an error at startup and ignored.

Setting	Required	Description

`ident`	yes	Unique name of the ruleset. Characters `A-Z a-z 0-9 _ -` only. If the name matches a built-in watch, the built-in settings are overridden.
`log`	yes	Absolute path of the log file to monitor. May be given multiple times. `*` matches within one directory, `**` matches recursively.
`regex`	yes	PHP regex *including delimiters*, matched against every new log line. It must capture a date and a host: use the placeholders `<REGEX:DATE>`, `<REGEX:HOST>` and `<REGEX:USER>`, or named groups `(?P<date>...)` and `(?P<host>...)`.
`scope`	yes	Services the IP is blocked for. One or more of `global`, `ssh`, `ftp`, `web`, `mail`, `dns`, comma-separated.
`alt_log`	no	Fallback log file, read when the file given in `log` is missing or unreadable.
`alt_regex`	no	Regex used instead of `regex` while `alt_log` is being read.
`ignore_regex`	no	Lines matching this regex are ignored even when `regex` matched.
`if_web`	no	`true` / `false`. If `true`, the ruleset is only active when `BANDAEMON_WWW_ROOT` is set. Default `false`.
`ignore_alltime`	no	`true` / `false`. If `true`, hits older than one hour do not contribute to the score. Use this for rules that should only react to bursts. Default `false`.
`ignore_previous_bans`	no	`true` / `false`. If `true`, earlier bans of the same IP do not raise the score. Default `false`.
`reduce`	no	Integer `0-20`. This many hits per minute are ignored before a score is calculated - useful for web logs where successful and failed logins cannot be told apart. Default `0`.
`bantime_factor`	no	Float `0.1-10`. The calculated score is multiplied by this value, which affects both whether an IP is banned at all and how long the ban lasts. Default `1`.

A fully commented reference file is installed as `conf.d/100-user.conf.sample`.

How BanDaemon Decides to Ban

BanDaemon does **not** use a fixed "N failures within T seconds" rule. Every match raises a score for the offending IP, and that score decides both whether the IP is banned and for how long:

- recent hits weigh far more than older ones (1 min / 5 min / 10 min / 1 hour windows, plus an all-time counter),
- previous bans of the same IP raise the score,
- the country of origin can raise or lower the score (see `BANDAEMON_COUNTRY_SCORES`),
- the result is multiplied by the ruleset's `bantime_factor`.

An IP is banned as soon as the score passes the internal threshold. The shortest ban is 10 minutes; a persistently abusive IP can be banned for up to a month. There is deliberately no per-ruleset hit threshold or ban time: use `bantime_factor` to make a ruleset more or less aggressive, `reduce` to grant a tolerance before scoring starts, and `ignore_alltime` / `ignore_previous_bans` to make a ruleset react to bursts only.

Fixed ban durations are available for **manual** bans only, via `ispp_bandaemon --ban=<ip> --bantime=<time>`.

Firewall Integration

iptables Configuration

Choose the appropriate table and mode:

- **PREROUTING** (recommended): Blocks traffic before routing decisions
- **INPUT**: Blocks traffic at the input chain level
- **INSERT**: Adds rules at the beginning of the chain
- **APPEND**: Adds rules at the end of the chain

iptables Legacy Mode

For systems with nf_tables compatibility issues:

```
// Enable legacy iptables commands
define('IPTABLES_LEGACY', true);
```

When to use legacy mode: - Getting "nf_tables incompatible" errors - Mixed environments with legacy and nf_tables rules - Container environments requiring legacy iptables - Older systems with legacy netfilter backend

Default: `false` (uses regular iptables commands)

Compatibility with Other Firewalls

ISPProtect BanDaemon works alongside: - **fail2ban**: Can complement each other - **UFW**: Works with Ubuntu's Uncomplicated Firewall - **firewalld**: Compatible with CentOS/RHEL firewall manager - **Cloud Security Groups**: AWS, Azure, GCP integration possible

---

Command Line Usage

Basic Commands

```
# Start the daemon
ispp_bandaemon

# Show help and available options
ispp_bandaemon --help

# Generate and view reports
ispp_bandaemon --report --screen
ispp_bandaemon --report --email=admin@example.com

# Manual IP management
ispp_bandaemon --ban=192.168.1.100 --bantime=1H
ispp_bandaemon --unban=192.168.1.100
ispp_bandaemon --showbans
ispp_bandaemon --showbans --ip=192.168.1.100
```

IP Management Commands

Banning IPs

```
# Ban an IP for 1 hour
ispp_bandaemon --ban=192.168.1.100 --bantime=1H

# Ban an IP for 30 minutes
ispp_bandaemon --ban=10.0.0.50 --bantime=30M

# Ban an IP for 1 day
ispp_bandaemon --ban=203.0.113.10 --bantime=1D

# Ban an IP for 3600 seconds (1 hour)
ispp_bandaemon --ban=198.51.100.20 --bantime=3600S
```

Unbanning IPs

```
# Unban a specific IP
ispp_bandaemon --unban=192.168.1.100

# Unban all known search engine bots
ispp_bandaemon --unbanbots
```

Viewing Bans

```
# Show all current bans
ispp_bandaemon --showbans

# Show bans for a specific IP
ispp_bandaemon --showbans --ip=192.168.1.100
```

Whitelist Management

```
# Add IP to whitelist
ispp_bandaemon --whitelist=192.168.1.10

# Remove IP from whitelist
ispp_bandaemon --whitelist=192.168.1.10 --remove

# Add subnet to whitelist
ispp_bandaemon --whitelist=192.168.1.0/24
```

Reporting and Analysis

```
# Generate daily report on screen
ispp_bandaemon --report --screen
```

```
# Generate today's report
ispp_bandaemon --report --screen --today

# Email report to multiple recipients
ispp_bandaemon --report --email=admin@example.com,security@example.com

# Generate DNS blacklist files
ispp_bandaemon --rbl

# Send abuse report for evidence file
ispp_bandaemon --abuse evidence/192.168.1.100_evidence.txt
```

Service Management

Systemd Commands (Modern Systems)

```
# Start the service
sudo systemctl start ispprotect_bandaemon

# Stop the service
sudo systemctl stop ispprotect_bandaemon

# Restart the service
sudo systemctl restart ispprotect_bandaemon

# Check service status
sudo systemctl status ispprotect_bandaemon

# Enable auto-start at boot
sudo systemctl enable ispprotect_bandaemon

# Disable auto-start at boot
sudo systemctl disable ispprotect_bandaemon

# View real-time logs
sudo journalctl -u ispprotect_bandaemon -f

# View recent logs
sudo journalctl -u ispprotect_bandaemon --since "1 hour ago"
```

Init.d Commands (Legacy Systems)

```
# Start the service
sudo service ispprotect_bandaemon start

# Stop the service
sudo service ispprotect_bandaemon stop

# Restart the service
sudo service ispprotect_bandaemon restart

# Check service status
sudo service ispprotect_bandaemon status

# View logs
sudo tail -f /var/log/ispprotect_bandaemon.log
```

Service Helper Script

Use the enhanced service management script:

```
# Navigate to installation directory
cd /opt/ispprotect_bandaemon
```

```
# Available commands
./systemd_service_helper.sh start
./systemd_service_helper.sh stop
./systemd_service_helper.sh restart
./systemd_service_helper.sh status
./systemd_service_helper.sh logs
./systemd_service_helper.sh health
./systemd_service_helper.sh migrate
```

Monitoring & Reporting

Real-time Monitoring

Log File Monitoring

```
# Monitor daemon logs in real-time
sudo journalctl -u ispprotect_bandaemon -f

# Monitor with specific log level
sudo journalctl -u ispprotect_bandaemon -p info -f

# Monitor log file directly
sudo tail -f /var/log/ispprotect_bandaemon.log
```

System Status Checks

```
# Check if daemon is running
sudo systemctl is-active ispprotect_bandaemon

# Check daemon health
./systemd_service_helper.sh health

# View current iptables rules (chains are in mangle table)
sudo iptables -t mangle -L -v -n

# If you get "nf_tables incompatible" errors, use legacy commands:
sudo iptables-legacy -t mangle -L -v -n
```

Automated Reporting

Daily Reports

Configure automatic daily reports in your configuration:

```
define('BANDAEMON_DAILY_REPORT', 'admin@example.com,security@example.com');
```

Set up a cron job for daily reports:

```
# Edit crontab
sudo crontab -e

# Add daily report at 6 AM
0 6 * * * /usr/local/bin/ispp_bandaemon --report --email=admin@example.com
```

Custom Report Scheduling

```
# Weekly summary report
0 6 * * 1 /usr/local/bin/ispp_bandaemon --report --email=weekly@example.com

# Monthly detailed report
0 6 1 * * /usr/local/bin/ispp_bandaemon --report --email=monthly@example.com --today
```

Performance Monitoring

Database Statistics

```
-- Connect to the database
mysql -u ispp_bandaemon -p ispprotect_bandaemon

-- Check ban statistics
SELECT COUNT(*) as total_bans,
       COUNT(CASE WHEN active=1 THEN 1 END) as active_bans,
       COUNT(CASE WHEN unban > NOW() THEN 1 END) as current_bans
FROM bans;

-- Top attacking countries
SELECT country, COUNT(*) as attacks
FROM bans
WHERE ban >= DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY country
ORDER BY attacks DESC
LIMIT 10;

-- Most targeted services
SELECT service, COUNT(*) as attacks
FROM auth_fails
WHERE authtime >= DATE_SUB(NOW(), INTERVAL 24 HOUR)
GROUP BY service
ORDER BY attacks DESC;
```

System Resource Usage

```
# Check memory usage
ps aux | grep ispp_bandaemon

# Check CPU usage
top -p $(pgrep -f ispp_bandaemon)

# Check disk usage
du -sh /opt/ispprotect_bandaemon/
```

IP Management

Understanding Ban Levels

ISPProtect BanDaemon uses a scoring system to determine ban duration:

- **Score 50-99**: 10-60 minutes
- **Score 100-174**: 1-12 hours
- **Score 175-249**: 12 hours - 1 week
- **Score 250-399**: 1-4 weeks
- **Score 400+**: 1 month (maximum ban)

Manual IP Management

Emergency Unban

If you accidentally ban yourself:

```
# From another server or console access
# Remove from specific service chain (replace SERVICE with: global, web, ssh, ftp, mail,
        or dns)
sudo iptables -D ispprotect-SERVICE-ban -s YOUR_IP -j DROP
```

```
sudo ip6tables -D ispprotect-SERVICE-ban -s YOUR_IPV6 -j DROP

# Then unban through the system
ispp_bandaemon --unban=YOUR_IP
```

Bulk Operations

```
# Unban multiple IPs
for ip in 192.168.1.100 192.168.1.101 192.168.1.102; do
    ispp_bandaemon --unban=$ip
done

# Add multiple IPs to whitelist
for ip in 192.168.1.10 192.168.1.11 192.168.1.12; do
    ispp_bandaemon --whitelist=$ip
done
```

Whitelist Management

Recommended Whitelist Entries

```
# Your management IPs
ispp_bandaemon --whitelist=YOUR_OFFICE_IP
ispp_bandaemon --whitelist=YOUR_HOME_IP

# Local network ranges
ispp_bandaemon --whitelist=192.168.1.0/24
ispp_bandaemon --whitelist=10.0.0.0/8

# Known good services
ispp_bandaemon --whitelist=MONITORING_SERVER_IP
ispp_bandaemon --whitelist=BACKUP_SERVER_IP
```

Search Engine Bots

The system automatically recognizes major search engine bots: - Google (googlebot) - Bing (bingbot) - Yahoo (slurp) - Yandex (yandex) - Baidu (baiduspider)

Use `--unbanbots` to release any accidentally banned crawlers.

Advanced Features

Evidence Collection

ISPProtect BanDaemon automatically collects evidence of attacks:

Viewing Evidence

```
# Navigate to evidence directory
cd /opt/ispprotect_bandaemon/evidence

# List evidence files
ls -la

# View evidence for a specific IP
cat 192.168.1.100_evidence.txt
```

Evidence File Format

Evidence files contain: - Timestamp of the attack - Source IP and geolocation - Service

targeted - Log entries showing the attack pattern - Whois information for the attacking IP

Abuse Reporting

Automatic Abuse Reports

Evidence collection and abuse contact lookup are configured in `config.inc.php`:

```
define('BANDAEMON_EVIDENCE', true);       // Write an evidence file per banned IP
define('BANDAEMON_EVIDENCE_SCORE', 100);  // Only for IPs reaching this score
define('BANDAEMON_ABUSE_ADDRESS', true);  // Look up the abuse contact via whois
define('EMAIL_SENDER', 'security@example.com');
```

Evidence files are written to `/opt/ispprotect_bandaemon/evidence/`. Sending the report is a separate, deliberate step - see *Manual Abuse Reports* below. BanDaemon never mails an abuse report on its own.

Manual Abuse Reports

```
# Send abuse report for collected evidence
ispp_bandaemon --abuse evidence/192.168.1.100_evidence.txt
```

GeoIP Integration

Country-based Filtering

BanDaemon looks up the country of every offending IP and adds a country specific value to its score. A few countries carry a built-in surcharge. You can override those values, add your own, or *lower* the score for countries you do business with, using `BANDAEMON_COUNTRY_SCORES` in `config.inc.php`:

```
// Comma-separated list of COUNTRYCODE:SCORE pairs.
// Positive values make a ban more likely and longer, negative values less likely.
define('BANDAEMON_COUNTRY_SCORES', 'CN:30,RU:30,VN:20,DE:-10');
```

Country codes are two-letter ISO codes; scores are whole numbers and may be negative. An entry replaces the built-in value for that country. For reference, an IP is banned once its total score passes 50.

Country scores are applied globally - they cannot be set per ruleset, and there is no per-country multiplier.

GeoIP Database Updates

```
# Update GeoIP databases (if using geoip-database-contrib)
sudo apt update && sudo apt upgrade geoip-database-contrib

# Or download manually
sudo wget -O /usr/share/GeoIP/GeoIP.dat
        http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz
sudo gunzip /usr/share/GeoIP/GeoIP.dat.gz
```

Distributed Operation

Multi-Server Setup

Configure multiple servers to share ban information:

1. **Shared Database**: Use a central MySQL/MariaDB server
2. **Server Identification**: Set unique `SERVER_IDENT` in each config

3. **Network Access**: Ensure database connectivity between servers

```php
// Server A configuration
define('DB_HOST', 'central-db.example.com');
define('SERVER_IDENT', 'web-server-01');

// Server B configuration
define('DB_HOST', 'central-db.example.com');
define('SERVER_IDENT', 'mail-server-01');
```

Load Balancer / Reverse Proxy Setups

BanDaemon reads log files; it does not see HTTP requests and therefore cannot evaluate `X-Forwarded-For` itself. Behind a load balancer or reverse proxy, all web hits would otherwise appear to come from the proxy.

Configure the **web server** to log the real client IP, then BanDaemon picks it up automatically:

```
# Apache - mod_remoteip
RemoteIPHeader X-Forwarded-For
RemoteIPInternalProxy 10.0.0.1 10.0.0.2

# nginx - realip module
set_real_ip_from 10.0.0.1;
set_real_ip_from 10.0.0.2;
real_ip_header X-Forwarded-For;
```

Also add the proxy addresses to `WHITELISTED_IPS` in `config.inc.php` so the load balancer itself can never be banned:

```php
define('WHITELISTED_IPS', '10.0.0.1,10.0.0.2');
```

Custom Service Integration

Creating Custom Monitors

Create a new configuration file in `conf.d/`, for example `conf.d/110-custom-app.conf`:

```
ident = custom-app
log = /var/log/myapp/security.log

# the <REGEX:DATE> placeholder expects the syslog format "Aug 21 09:15:42".
# for other timestamp formats use your own named group, as shown here:
regex = /^(?P<date>\d{4}-\d{2}-\d{2}\s+\d{2}:\d{2}:\d{2})\s+SECURITY_VIOLATION:\s+IP=
<REGEX:HOST>\s/

scope = global

# react to bursts only - ignore hits older than one hour
ignore_alltime = true

# this log only contains confirmed attacks, so score aggressively
bantime_factor = 2
```

Because every line in such a log is a confirmed attack, leave `reduce` at `0` and raise `bantime_factor` instead - that makes a single hit weigh far more and gets the IP banned much sooner.

API Integration

For applications with APIs, create log entries that BanDaemon can monitor. The format below matches the `regex` in the example above:

```php
// In your application
function logSecurityViolation($ip, $violation_type) {
    $log_entry = date('Y-m-d H:i:s') . " SECURITY_VIOLATION: IP=$ip
        TYPE=$violation_type\n";
    file_put_contents('/var/log/myapp/security.log', $log_entry, FILE_APPEND);
}
```

Every regex must capture both a date and a host. Test a new pattern before relying on it:

```
# append a test line and watch the daemon react
echo "$(date '+%Y-%m-%d %H:%M:%S') SECURITY_VIOLATION: IP=203.0.113.10 TYPE=test" \
  >> /var/log/myapp/security.log

journalctl -u ispprotect_bandaemon -f
```

Set `BANDAEMON_TESTMODE` to `true` while testing so matches are detected and logged without any iptables rules being written.

Troubleshooting

Common Issues

Service Won't Start

```
# Check service status
sudo systemctl status ispprotect_bandaemon

# Check for configuration errors
sudo journalctl -u ispprotect_bandaemon --since "5 minutes ago"

# Verify PHP syntax
php -l /opt/ispprotect_bandaemon/ispp_bandaemon.php

# Check file permissions
ls -la /opt/ispprotect_bandaemon/
```

Database Connection Issues

```
# Test database connection
mysql -h localhost -u ispp_bandaemon -p ispprotect_bandaemon

# Check MySQL service
sudo systemctl status mysql
sudo systemctl status mariadb

# Verify database exists
mysql -u root -p -e "SHOW DATABASES LIKE 'ispprotect_bandaemon';"
```

iptables Problems

```
# Check if iptables is installed
which iptables

# Verify iptables rules (use iptables-legacy if nf_tables errors occur)
sudo iptables -t mangle -L -v -n

# Check for conflicting rules
sudo iptables -t mangle -L -v -n | grep -i ispprotect

# Reset iptables rules (CAUTION!)
sudo iptables -t mangle -F ispprotect-global-ban
sudo iptables -t mangle -F ispprotect-web-ban
```

```
sudo iptables -t mangle -F ispprotect-ssh-ban
sudo iptables -t mangle -F ispprotect-ftp-ban
sudo iptables -t mangle -F ispprotect-mail-ban
sudo iptables -t mangle -F ispprotect-dns-ban
sudo iptables -F ispprotect-web-ban
sudo iptables -F ispprotect-ssh-ban
sudo iptables -F ispprotect-ftp-ban
sudo iptables -F ispprotect-mail-ban
sudo iptables -F ispprotect-dns-ban
sudo ip6tables -F ispprotect-global-ban
sudo ip6tables -F ispprotect-web-ban
sudo ip6tables -F ispprotect-ssh-ban
sudo ip6tables -F ispprotect-ftp-ban
sudo ip6tables -F ispprotect-mail-ban
sudo ip6tables -F ispprotect-dns-ban
```

Custom conf.d Ruleset Is Not Working

Check the daemon log directly after a restart:

```
sudo systemctl restart ispprotect_bandaemon
sudo journalctl -u ispprotect_bandaemon -n 100 | grep -iE "config file|invalid
        setting|missing ident"
```

Message	Cause	Fix
`section headers like [name] are not supported`	The file uses INI/fail2ban syntax.	Replace `[name]` with `ident = name`.
`invalid setting <key>`	The key does not exist in BanDaemon.	Remove it. See *Available Configuration Options*. The daemon names the correct setting where one exists.
`Missing ident line in config file ...`	No `ident = ...` line, so **nothing** from this file is loaded and the log file is not monitored.	Add an `ident` line.
`Missing line in config file ... (setting log)`	`log` is missing - often because `logfile` was used.	Rename `logfile` to `log`.
`unsupported setting(s) were ignored`	The ruleset **is** active, but the listed keys do nothing.	Remove them and tune with `reduce` / `bantime_factor` instead.

A ruleset is only in use when you see `Read config file ... and added <ident> ruleset.`

If the ruleset loads but never triggers, the regex is the usual culprit - it must match the whole line and capture both a date and a host. Enable `BANDAEMON_VERBOSE` and inject a test line to see whether the daemon registers a hit.

Permission Errors

```
# Fix file permissions
sudo chown -R root:root /opt/ispprotect_bandaemon/
sudo chmod 755 /opt/ispprotect_bandaemon/ispp_bandaemon
sudo chmod 644 /opt/ispprotect_bandaemon/*.php

# Fix log file permissions
sudo touch /var/log/ispprotect_bandaemon.log
sudo chown root:root /var/log/ispprotect_bandaemon.log
sudo chmod 644 /var/log/ispprotect_bandaemon.log
```

PID File Race Condition

If you see this error in systemd logs:

```
systemd[1]: ispprotect_bandaemon.service: Can't open PID file
/run/ispprotect_bandaemon.pid (yet?) after start: No such file or directory
```

This is a race condition where systemd checks for the PID file before the daemon creates it. The service still works correctly, but you can eliminate this warning by adding a timeout setting.

Solution:

1. **Edit the systemd service file:**

   ```
   sudo nano /etc/systemd/system/ispprotect_bandaemon.service
   ```

2. **For modern systemd (v229+)**, add this line in the `[Service]` section after the `PIDFile=` line:

   ```
   PIDFileTimeoutSec=10
   ```

   For older systemd versions, increase the startup timeout instead:

   ```
   TimeoutStartSec=60
   ```

3. **Reload systemd and restart the service:**

   ```
   sudo systemctl daemon-reload
   sudo systemctl restart ispprotect_bandaemon
   ```

This gives systemd more time to wait for the PID file to appear, eliminating the race condition.

Note: If you get an error like “Unknown lvalue ‘PIDFileTimeoutSec’”, your systemd version is older and doesn’t support this directive. Use the `TimeoutStartSec=60` approach instead.

Debug Mode

Enable Verbose Logging

Edit `/opt/ispprotect_bandaemon/config.inc.php`:

```
define('BANDAEMON_VERBOSE', true);
define('BANDAEMON_LOGNAME', true);
define('BANDAEMON_TESTMODE', true); // Test mode - no actual bans
```

Test Mode

Run in test mode to see what would be banned without actually banning:

```
define('BANDAEMON_TESTMODE', true);
```

Manual Testing

```
# Test specific log entries
echo "$(date) Failed password for root from 192.168.1.100 port 22 ssh2" >>
        /var/log/auth.log

# Monitor the response
sudo journalctl -u ispprotect_bandaemon -f
```

Log Analysis

Common Log Patterns

```
# Search for ban events
sudo grep "BANNED" /var/log/ispprotect_bandaemon.log

# Search for unban events
sudo grep "UNBANNED" /var/log/ispprotect_bandaemon.log

# Search for errors
sudo grep "ERROR" /var/log/ispprotect_bandaemon.log

# Search for specific IP
sudo grep "192.168.1.100" /var/log/ispprotect_bandaemon.log
```

Performance Analysis

```
# Check processing speed
sudo grep "Processing time" /var/log/ispprotect_bandaemon.log

# Monitor memory usage
sudo grep "Memory usage" /var/log/ispprotect_bandaemon.log

# Check database query performance
sudo grep "Database query" /var/log/ispprotect_bandaemon.log
```

Recovery Procedures

Emergency Stop

```
# Stop the service immediately
sudo systemctl stop ispprotect_bandaemon

# Clear all ban rules (CAUTION!)
sudo iptables -F ispprotect-global-ban
sudo iptables -F ispprotect-web-ban
sudo iptables -F ispprotect-ssh-ban
sudo iptables -F ispprotect-ftp-ban
sudo iptables -F ispprotect-mail-ban
sudo iptables -F ispprotect-dns-ban
sudo ip6tables -F ispprotect-global-ban
sudo ip6tables -F ispprotect-web-ban
sudo ip6tables -F ispprotect-ssh-ban
sudo ip6tables -F ispprotect-ftp-ban
sudo ip6tables -F ispprotect-mail-ban
sudo ip6tables -F ispprotect-dns-ban
```

Configuration Reset

```
# Backup current configuration
sudo cp /opt/ispprotect_bandaemon/config.inc.php /tmp/config.backup

# Reset to default configuration
sudo cp /opt/ispprotect_bandaemon/config.inc.php.sample
        /opt/ispprotect_bandaemon/config.inc.php

# Edit with your settings
sudo nano /opt/ispprotect_bandaemon/config.inc.php
```

Database Recovery

```
# Backup database
mysqldump -u ispp_bandaemon -p ispprotect_bandaemon > bandaemon_backup.sql

# Clear all bans (if needed)
mysql -u ispp_bandaemon -p ispprotect_bandaemon -e "UPDATE bans SET active=0;"
```

```
# Restore from backup
mysql -u ispp_bandaemon -p ispprotect_bandaemon < bandaemon_backup.sql
```

Best Practices

Security Recommendations

Whitelist Management

1. **Always whitelist your management IPs** before enabling the service
2. **Use subnet ranges** for office networks rather than individual IPs
3. **Regularly review** your whitelist for outdated entries
4. **Document** all whitelist entries with reasons

Configuration Security

```
# Secure configuration file permissions
sudo chmod 600 /opt/ispprotect_bandaemon/config.inc.php
sudo chown root:root /opt/ispprotect_bandaemon/config.inc.php

# Use strong database passwords
# Consider using database SSL connections for remote databases
```

Monitoring Setup

1. **Enable email reports** for daily summaries
2. **Set up log rotation** to prevent disk space issues
3. **Monitor system resources** regularly
4. **Create alerts** for service failures

Performance Optimization

Database Optimization

```
-- Add indexes for better performance
ALTER TABLE auth_fails ADD INDEX idx_ip_service_time (ip, service, authtime);
ALTER TABLE bans ADD INDEX idx_active_unban (active, unban);

-- Regular maintenance
OPTIMIZE TABLE auth_fails;
OPTIMIZE TABLE bans;
```

Log File Management

```
# Configure logrotate for daemon logs
sudo nano /etc/logrotate.d/ispprotect_bandaemon

# Add configuration:
/var/log/ispprotect_bandaemon.log {
    daily
    rotate 30
    compress
    delaycompress
    missingok
    notifempty
    postrotate
        systemctl reload ispprotect_bandaemon
    endscript
}
```

System Tuning

```
# Increase file descriptor limits if monitoring many log files
echo "fs.file-max = 65536" >> /etc/sysctl.conf

# Optimize MySQL for BanDaemon
# Add to /etc/mysql/mysql.conf.d/bandaemon.cnf:
[mysqld]
innodb_buffer_pool_size = 128M
query_cache_size = 32M
query_cache_limit = 2M
```

Maintenance Procedures

Regular Tasks

1. **Weekly**: Review ban statistics and adjust `reduce` / `bantime_factor`
2. **Monthly**: Clean old evidence files and optimize database
3. **Quarterly**: Update GeoIP databases and review configurations
4. **Annually**: Review and update whitelist entries

Backup Procedures

```
# Create backup script
#!/bin/bash
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/backup/ispprotect_$DATE"

mkdir -p $BACKUP_DIR

# Backup configuration
cp -r /opt/ispprotect_bandaemon/ $BACKUP_DIR/

# Backup database
mysqldump -u ispp_bandaemon -p ispprotect_bandaemon > $BACKUP_DIR/database.sql

# Compress backup
tar -czf /backup/ispprotect_backup_$DATE.tar.gz $BACKUP_DIR/
rm -rf $BACKUP_DIR/
```

Update Procedures

```
# Before updating
sudo systemctl stop ispprotect_bandaemon
sudo cp -r /opt/ispprotect_bandaemon/ /backup/ispprotect_pre_update/

# After updating
sudo systemctl start ispprotect_bandaemon
sudo systemctl status ispprotect_bandaemon
```

Integration Guidelines

Web Server Integration

For Apache/Nginx integration:

```
# Apache .htaccess example
<RequireAll>
    Require all granted
    Require not ip 192.168.1.100  # Banned IP
</RequireAll>
```

Monitoring Integration

```
# Nagios/Icinga check
#!/bin/bash
if systemctl is-active --quiet ispprotect_bandaemon; then
    echo "OK - ISPProtect BanDaemon is running"
    exit 0
else
    echo "CRITICAL - ISPProtect BanDaemon is not running"
    exit 2
fi
```

Backup Integration

```
# Include in your backup scripts
tar -czf server_backup.tar.gz \
    /etc/ \
    /opt/ispprotect_bandaemon/ \
    --exclude=/opt/ispprotect_bandaemon/evidence/
```

Support and Resources

Getting Help

- **Log Files**: Always check `/var/log/ispprotect_bandaemon.log` first
- **Service Status**: Use `systemctl status ispprotect_bandaemon`
- **Configuration**: Verify `/opt/ispprotect_bandaemon/config.inc.php`
- **Database**: Check database connectivity and table structure

Useful Commands Reference

```
# Quick status check (use iptables-legacy if nf_tables errors occur)
sudo systemctl status ispprotect_bandaemon && \
sudo iptables -t mangle -L -v -n --line-numbers && \
echo "To remove a specific rule, use:" && \
echo "sudo iptables -t mangle -D ispprotect-SERVICE-ban -s YOUR_IP -j DROP" && \
echo "If nf_tables errors occur, replace 'iptables' with 'iptables-legacy'"

# View recent activity
sudo journalctl -u ispprotect_bandaemon --since "1 hour ago" | tail -20

# Check configuration syntax
php -l /opt/ispprotect_bandaemon/config.inc.php

# Verify that all conf.d rulesets were loaded (no "invalid setting" / "Missing ident")
journalctl -u ispprotect_bandaemon | grep -iE "config file|invalid setting|missing ident"
```

Version Information

- **Current Version**: 2.1.1
- **PHP Compatibility**: 5.6 through 8.x
- **Database Support**: MySQL 5.5+, MariaDB 10.0+
- **OS Support**: Linux with iptables

This documentation covers ISPProtect BanDaemon 2.1.1. For the latest updates and additional resources, please visit the official website.