

ISPPROTECT

# Malware-Scanner Handbuch für Anwender

Ihr Server-Detektiv.

Version 1.0.16
Projektfarm GmbH · ispprotect.com

# Inhalt


1. Einführung 03

2. Installation 04

3. Lizenzierung 06

4. Schnellstart 07

5. Referenz der Kommandozeile 08

6. Ablauf eines Scans 13

7. Erkennung von CMS, Plugins und Versionen 15

8. Signatur-Feeds 16

9. Datenbank-Scan 17

10. E-Mail-Berichte 18

11. Konfigurationsassistent 19

12. Performance und Tuning 20

13. Ergebnisse, Quarantäne und Whitelist 21

14. Proben einreichen 22

15. ISPProtect Console (malware-agent) 23

16. Daten, Zustand und Umgebung 27

17. Der ClamAV-Verdict-Dienst 29

18. Fehlerbehebung 30

19. Support und Rechtliches 31

# 1. Einführung

ISPProtect ist ein Malware-Erkennungssystem für Linux-Webhosting-Umgebungen. Es durchsucht Web-Roots, Anwendungen und Datenbanken nach Malware, Webshells, Spam-Tools sowie verwundbarer oder veralteter Software — und meldet die Funde, bevor Ihre Nutzer oder Provider darauf aufmerksam werden.

Ein Scan kombiniert drei voneinander unabhängige Erkennungsebenen:

| Ebene | Engine | Was sie erkennt |
|---|---|---|
| 1 | Engine für binäre Signaturen (ClamAV-Verdict-Dienst) | Bekannte Malware, Backdoors und Trojaner anhand binärer Signaturen |
| 2 | Muster- und Heuristik-Engine | 184 mitgelieferte Musterregeln, Verschleierungsketten, verdächtige Code-Konstrukte |
| 3 | Bayes-Klassifikator (optional) | Bisher unbekannte Malware, angelernt aus verifizierter Malware und sauberen Korpora |

Jeder Lauf schreibt einen vollständigen JSON-Bericht sowie Textartefakte für die ausgeführten Phasen und kann die Ergebnisse per E-Mail zustellen. Bestätigte Treffer lassen sich in Quarantäne verschieben und wiederherstellen.

## 1.1 Architektur im Überblick

Der Scanner selbst ist ein einzelnes, eigenständiges Go-Binary (`malware-scan`). Das Scannen nach binären Signaturen übernimmt ein separater kleiner Dienst, `clamav-ipc-service`, der `libclamav` einbindet. Der Scanner startet ihn als Kindprozess, fragt über einen lokalen Socket Prüfergebnisse ab und beendet ihn, wenn der Scan endet. Fehlt der Dienst — bei portablen Installationen oder einem bewusst entfernten Paket —, meldet der Scanner die Binär-Engine als nicht verfügbar und setzt den Scan mit Muster-, Bayes- und Datenbank-Scan fort.

# 2. Installation

## 2.1 Aus dem APT-Repository (empfohlen)

Die Pakete sind signiert und werden pro Distribution veröffentlicht. Wählen Sie die Suite, die zu Ihrem System passt — wer die Pakete einer anderen Distribution installiert, erhält ein `clamav-ipc-service`-Binary, das gegen eine andere glibc gebaut wurde.

```
# prerequisites
sudo apt update
sudo apt install -y ca-certificates curl gnupg

# add and trust the repository key
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://repo.ispprotect.com/public.key | \
    sudo gpg --batch --yes --dearmor -o /etc/apt/keyrings/ispprotect.gpg

# add the repository for your distribution
. /etc/os-release
echo "deb [signed-by=/etc/apt/keyrings/ispprotect.gpg] https://repo.ispprotect.com/ $VERSION_CODENAME main" |
\
    sudo tee /etc/apt/sources.list.d/malware-scan.list

# install
sudo apt update
sudo apt install malware-scan
```

Debian 11 (Bullseye) hat im August 2026 den LTS-Support verlassen, seine Pakete wurden nach archive.debian.org verschoben. Meldet `apt` bei der Installation der Abhängigkeiten 404-Fehler, stellen Sie die Paketquellen des Systems zuerst auf das Archiv um.

Unter AlmaLinux, Rocky Linux, RHEL und anderen EL 8/9/10 (x86_64), wobei EPEL `clamav-lib` bereitstellt:

```
# EPEL provides clamav-lib
sudo dnf install -y epel-release

# Repository definition for this EL release (el8, el9 or el10)
sudo curl -fsSL -o /etc/yum.repos.d/ispprotect.repo \
    https://repo.ispprotect.com/rpm/ispprotect-el$(rpm -E %rhel).repo

# Import the signing key
sudo rpm --import https://repo.ispprotect.com/public.key

sudo dnf install malware-scan
```

**Signaturschlüssel.** Pakete und Repository-Metadaten sind mit folgendem Schlüssel signiert:

```
2FCF BFB8 26F4 962B A04A  EB1B 6022 3BD3 F8E6 4310
ISPProtect Repository Signing Key 2026 <info@ispprotect.com>, valid until 2029-10-10
```

Prüfen Sie ihn mit `curl -fsSL https://repo.ispprotect.com/public.key | gpg --show-keys --with-fingerprint`.

Unterstützte Distributionen: Debian 10 (Legacy), 11, 12, 13, Ubuntu 22.04, 24.04 sowie EL 8, 9, 10. `malware-scan` zieht `clamav-ipc-service` als Abhängigkeit nach; dafür muss nichts konfiguriert werden.

Updates kommen über denselben Kanal — mit `apt upgrade` bzw. `dnf upgrade` bleibt der Scanner aktuell. Installation und Updates des Scanners erfolgen bewusst **ausschließlich** über signierte Systempakete.

### 2.2 Portables Paket

Ein portables Tarball läuft ohne Installation. Ist `clamav-ipc-service` nicht vorhanden, wird die Binär-Engine als nicht verfügbar gemeldet, und der Scan läuft mit den übrigen Engines weiter.

# 3. Lizenzierung

| Lizenz | Preis | Hinweise |
|---|---|---|
| Testversion | kostenlos | Lizenzschlüssel: `TRIAL`. Keine Registrierung. Verfügbarkeit, Ablauf und Scan-Limits werden vom Lizenzserver geprüft. |
| Jahreslizenz | 82,68 € / Jahr (6,89 € / Monat) | Ein Server, unbegrenzte Scans, Zusatzmodul BanDaemon zur IP-Sperrung inklusive. |
| Pro Scan | 5 € / 8 € / 20 € / 35 € / 60 € | 5 / 10 / 25 / 50 / 100 Scans. Nicht serverbezogen, ohne zeitliche Begrenzung. |

Alle Preise verstehen sich zuzüglich Umsatzsteuer; für EU-Kunden ohne gültige USt-IdNr. fällt Umsatzsteuer an. Aktuelle Preise und Angebote: ispprotect.com.

Übergeben Sie Ihre Lizenz entweder direkt oder als Pfad zu einer Schlüsseldatei:

```
malware-scan --license-key TRIAL --path /var/www          # trial
malware-scan --license-key /etc/malware-scan/license.key  # key file
```

Um den Schlüssel aus der Prozessliste herauszuhalten, übergeben Sie ihn stattdessen über die Umgebung. Der Scanner verwendet die erste verfügbare Quelle aus `--license-key`, `MALWARE_SCAN_LICENSE_KEY`, der in `MALWARE_SCAN_LICENSE_FILE` genannten Datei und dem gespeicherten Schlüssel und entfernt beide Variablen aus seiner Umgebung, sobald sie gelesen wurden:

```
MALWARE_SCAN_LICENSE_FILE=/etc/malware-scan/license.key malware-scan --path /var/www
```

`malware-scan --license-status` zeigt die Lizenz an; zusammen mit `--json-out` gibt es sie als JSON aus (`schema_version`, `valid`, `status`, `message`, `trial`, `plan`, `expires_at`, `expires_date`, `days_remaining`, `scans_remaining`, `unlimited_scans`, `max_servers`, `key_masked`). `status` ist `valid`, `expired`, `locked`, `invalid`, `no_scans` oder `error` (kein Lizenzserver erreichbar); Felder, die der Server nicht liefert, sind `null`. Der Befehl endet mit Exit-Code 0 bei gültiger Lizenz und andernfalls mit 3.

Die Verknüpfung eines Servers mit der ISPProtect Console (Abschnitt 15) erfordert eine kostenpflichtige Lizenz: Jahreslizenz oder Scan-Pakete. Testlizenzen lassen sich nicht verknüpfen.

# 4. Schnellstart

```
# basic scan of one site
malware-scan --path /path/to/website

# deep scan including the database check
malware-scan --deep-check --db-scan --path /var/www

# scan and email the results
malware-scan --email-results admin@example.com --path /var/www
```

Ohne `--license-key` fragt der Scanner nach einem Schlüssel. Ein einfacher `--path`-Scan führt alle Malware-Engines, die Versionsprüfungen für CMS und Plugins sowie die Besitzerprüfung aus (Abschnitt 6).

# 5. Referenz der Kommandozeile

```
Usage: malware-scan --path <path> [options]
```

| Option | Beschreibung |
|---|---|
| `--license-key <key>` | Lizenzschlüssel oder Pfad zu einer Schlüsseldatei |
| `--path <path>` | Scan-Wurzel; Globs in Anführungszeichen sind zulässig, wenn sie zu genau einer Wurzel aufgelöst werden |
| `--no-malware-scan` | Malware-Engines und jede Arbeit an Dateiinhalten überspringen |
| `--no-version-scan` | Versionsprüfungen für CMS und WordPress-Plugins überspringen |
| `--no-plugin-version-scan` | Nur die Versionsprüfung der WordPress-Plugins überspringen |
| `--version-scan <software>` | Nur ein unterstütztes CMS oder Produkt prüfen |
| `--bayes-scan` | Den Bayes-Klassifikator (ML) aktivieren |
| `--no-image-scan` | Den zusätzlichen binären Bilddurchlauf deaktivieren (standardmäßig aktiv) |
| `--deep-check` | Tiefenprüfung verdächtiger Dateien (langsamer, gründlicher) |
| `--no-deep-check-network` | Tiefenprüfung nur für lokale Includes: keine DNS-Abfragen, keine Downloads |
| `--deep-check-timeout <s>` | Sekunden pro Remote-Download in der Tiefenprüfung (Standard 5, Maximum 300) |
| `--allow-single` | Das Scannen einer einzelnen Datei erlauben |
| `--scan-level <1\|2>` | 1 = binäre Signaturen und Muster (Standard), 2 = nur Muster |
| `--regex-scan-all-files` | Die Musterregeln auf alle Dateitypen anwenden |
| `--check-fps[=false]` | Bekannte False-Positive-Treffer unterdrücken (Standard: an) |
| `--show-hits` | Treffer sofort ausgeben |
| `--show-progress` | Einen Fortschrittsbalken für den Datei-Scan anzeigen |
| `--incremental` | Persistenten Clean-/Malware-Hash-Zustand zwischen Läufen wiederverwenden |
| `--concurrency <n>` | Gleichzeitige Datei-Scans (Standard 5, Maximum 1024) |
| `--exclude <glob>` / `--include <glob>` | Pfadfilter; wiederholbar; `*` und `**` werden unterstützt |
| `--exclude-from <file>` / `--include-from <file>` | Filtermuster aus einer Datei laden; wiederholbar |
| `--max-age <days>` | Nur Dateien berücksichtigen, die jünger als diese Anzahl Tage sind |
| `--ignore-chmod0` | Dateien mit Modus 000 ignorieren |
| `--ignore <name>` | Eine Signatur nach ID oder Name unterdrücken; wiederholbar (Abschnitt 13) |
| `--fail-on <severity>` | Niedrigster Schweregrad, der für Exit-Code 1 zählt: `info`, `low`, `medium`, `high` oder `critical` (Standard `medium`) |

| Option | Beschreibung |
|---|---|
| `--no-owner-check` | Besitzerprüfung und Prüfung auf leere Skriptdateien überspringen (standardmäßig an) |
| `--verify-checksums` | WordPress-Core- und Plugin-Dateien mit den offiziellen Prüfsummen vergleichen (Abschnitt 6) |
| `--time-check` | Auf verdächtige Zeitstempel von Dateien prüfen |
| `--blacklist-check` | Domains aus der Webserver-Konfiguration gegen die DNS-Blacklists aus `blacklist.conf` prüfen (Abschnitt 10.1) |
| `--blacklist-config <file>` | DNS-Blacklist-Konfiguration (Standard: `blacklist.conf` im Konfigurationsverzeichnis) |
| `--db-scan` | MySQL-/MariaDB-Datenbanken scannen |
| `--db-scan-config <file>` | Datenbank-Konfigurationsdatei |
| `--db-socket <path\|auto>` | Verbindung über einen Unix-Socket; kein Passwort nötig |
| `--db-user <name>` | Datenbankbenutzer (Standard: konfigurierter Benutzer oder der OS-Benutzer) |
| `--db-scan-maxrows <n>` | Höchstzahl gelesener Zeilen pro Tabelle (Standard 1000000, 0 = alle) |
| `--db-exclude <pattern>` / `--db-only <pattern>` | Datenbanken ausschließen / nur diese Datenbanken scannen; wiederholbar |
| `--db-exclude-table <pattern>` / `--db-only-table <pattern>` | Tabellen ausschließen / nur diese Tabellen scannen; wiederholbar |
| `--db-no-context` | Treffer-Werte aus der Datenbank in den Berichten weglassen |
| `--email-results <address>` | Den Bericht an diese Adresse senden; wiederholbar |
| `--email-config <file>` | SMTP-Konfiguration für den Berichtsversand |
| `--email-sender <address>` | Absenderadresse der Berichts-E-Mail |
| `--email-hostname <name>` | In der Berichts-E-Mail verwendeter Hostname |
| `--email-empty-results` | Die E-Mail auch senden, wenn nichts gefunden wurde |
| `--quarantine <dir>` | Erkannte Malware in dieses Verzeichnis verschieben (schaltet den Quarantäne-Modus ein) |
| `--quarantine-mode` | Den Quarantäne-Modus einschalten (von `--quarantine` vorausgesetzt) |
| `--restore` | Dateien aus dem Quarantäne-Verzeichnis wiederherstellen |
| `--results-dir <dir>` | Berichtsverzeichnis (Standard `/var/log/malware-scan`) |
| `--json-file <file>` | Den vollständigen JSON-Laufbericht zusätzlich hierhin schreiben |
| `--json-out` | Den vollständigen JSON-Laufbericht auf stdout schreiben (erfordert `--non-interactive`) |
| `--non-interactive` | Nie nachfragen; von `--json-out` vorausgesetzt |
| `--scan-id <id>` | Diese Scan-ID verwenden (1–64 Zeichen aus `A-Z a-z 0-9 . _ -`) |

| Option | Beschreibung |
|---|---|
| `--status-file <file>` | Einzeiliger Status `PHASE;percent;hits;eta;scan_id` |
| `--progress-fd <n>` | JSON-Lines-Fortschrittsereignisse an den geerbten Deskriptor *n* |
| `--progress-file <file>` | JSON-Lines-Fortschrittsereignisse in eine Datei |
| `--license-status` | Lizenz anzeigen und beenden (JSON mit `--json-out`) |
| `--list-signatures` | Signaturen der Scan-Engine zeilenweise auflisten und beenden; braucht weder Lizenz noch Netzwerk |
| `--json` | Mit `--list-signatures`: die vollständigen Einträge als ein JSON-Array ausgeben; allein ist es ein Konfigurationsfehler |
| `--signature-dir <dir>` | Speicherort der heruntergeladenen Signatur-Feeds |
| `--no-sig-update` | Signatur-Feeds von Drittanbietern nie herunterladen |
| `--update-signatures` | Die Feeds auch dann aktualisieren, wenn sie aktuell sind |
| `--sig-update-interval <h>` | Stunden, die ein heruntergeladener Feed als aktuell gilt (Standard 24) |
| `--extended-signatures` | Zusätzlich die Mail- und Spam-Sets laden (mehr False Positives) |
| `--use-tmpfs` | `/dev/shm` für temporäre Dateien verwenden, sofern verfügbar |
| `--use-tmp <dir>` | Ein beschreibbares temporäres Verzeichnis verwenden |
| `--whitelist <file>` | Den SHA-256 einer Datei zur Whitelist hinzufügen und beenden |
| `--whitelist-path <file>` | Zu verwendende Whitelist-Datei (Standard `~/.ispp_scan.whitelist`) |
| `--configure [db\|email\|all]` | Interaktiver Konfigurationsassistent, danach beenden |
| `--configure-dir <dir>` | Wohin der Assistent seine Dateien schreibt |
| `--system-report` | Eine lokale `system_report.txt` schreiben und beenden |
| `--cleanup [days]` | Berichte löschen, die älter als diese Anzahl Tage sind (ohne Wert 7) |
| `--check-permissions [strict\|lax]` | Berechtigungsprüfung auch ohne Datei-Malware-Scan ausführen (ohne Wert `strict`) |
| `--perm-check` / `--perm-check-strict` | Berechtigungsprüfung im lockeren / strengen Modus, auch ohne Datei-Malware-Scan |
| `--no-check-permissions` | Berechtigungsprüfung abschalten; nicht kombinierbar mit den drei Optionen darüber |
| `--learn-ham <file>` / `--learn-spam <file>` | Das Bayes-Modell mit einer harmlosen / schädlichen Datei trainieren |
| `--reset-ip-address` | Die IP-Bindung der Lizenz zurücksetzen |
| `--debug` / `--debug-regex-matches` | Diagnoseausgabe / genaue Treffer der regulären Ausdrücke ausgeben (kann Dateiinhalte offenlegen) |
| `--version` | Die Scanner-Version anzeigen |
| `--timing` | Gesamtlaufzeit des Scans ausgeben |

| Option | Beschreibung |
|---|---|
| `--quiet` | Nur Terminalausgaben unterdrücken — Berichte werden weiterhin geschrieben |
| `--report` / `--false-positive` | Eine Probe einreichen: nicht erkannte Malware / False Positive |
| `--confirm-file-upload` | Ausdrückliche Zustimmung, die für jeden Proben-Upload erforderlich ist |
| `--email` | Optionale Kontaktadresse beim Einreichen von Proben |
| `--help` | Die vollständige Optionsliste anzeigen |

Die Schwachstellenprüfung der WordPress-Plugins läuft immer. Die Besitzerprüfung läuft zusammen mit der Dateisuche, sofern nicht `--no-owner-check` angegeben wird; für eine einzelne Datei läuft sie nicht. Die Berechtigungsprüfung läuft bei jedem Datei-Malware-Scan im lockeren Modus, sofern nicht `--no-check-permissions` angegeben wird. Die Bayes-Klassifikation ist aus, solange `--bayes-scan` nicht angegeben wird; der erweiterte Bild-Scan ist an, solange `--no-image-scan` nicht angegeben wird.

`--no-malware-scan` erlaubt weiterhin die davon unabhängigen Metadaten-Phasen für CMS und Plugins, initialisiert aber keine Malware-Scanner, bildet keine Hashes über das Dateiinventar, führt keine Tiefenprüfungen aus und befüllt den Malware-Cache nicht.

### 5.1 Exit-Codes

| Code | Bedeutung |
|---|---|
| 0 | Scan abgeschlossen, kein Fund auf oder über der Schwelle von `--fail-on` |
| 1 | Scan abgeschlossen, mindestens ein Fund auf oder über der Schwelle von `--fail-on` (Standard `medium`) |
| 2 | Konfigurations- oder Kommandozeilenfehler (auch eine wiederverwendete `--scan-id`) |
| 3 | Lizenz ungültig, abgelaufen, gesperrt, keine Scans mehr übrig oder nicht prüfbar |
| 4 | E/A- oder Laufzeitfehler: eine Prüfung ist fehlgeschlagen, das Ergebnisverzeichnis war nicht nutzbar oder der Bericht konnte nicht geschrieben werden |
| 130 | Durch SIGINT oder SIGTERM unterbrochen; ein Teilbericht wird geschrieben |

Ein Fund ist jede Erkennung in einer Datei, Remote-Ressource, einem Verzeichnis, einer Datenbank oder einer Domain auf der Blacklist; jeder hat einen Schweregrad `info`, `low`, `medium`, `high` oder `critical`. Nur Funde auf oder über der Schwelle von `--fail-on` (Standard `medium`) führen zu Exit-Code 1; schwächere Funde bleiben im Bericht. Ein Abbruch hat Vorrang vor Funden, Funde haben Vorrang vor fehlgeschlagenen Prüfungen, und fehlgeschlagene Prüfungen ergeben Exit-Code 4; sie stehen unter `errors` im Bericht. Veraltete Software, verwundbare Plugins und Berechtigungsprobleme werden gemeldet, zählen aber nicht als Funde. Ein zweites SIGINT/SIGTERM beendet den Scanner sofort und ohne Bericht.

### 5.2 Automatisierung

Für ein Control-Panel oder ein Skript kombinieren Sie `--non-interactive --json-out` mit einer festen `--scan-id` und einem Fortschrittsstrom:

```
malware-scan --path /var/www --non-interactive --scan-id panel-42 \
  --progress-fd 3 --json-out 3>progress.jsonl >report.json
```

Mit `--json-out` (oder `--progress-fd 1`) enthält stdout ausschließlich JSON; jede Meldung für einen Menschen geht an stderr.

**Bericht.** `scan_report_<scan-id>.json` (wird immer in das Ergebnisverzeichnis geschrieben, mit `--json-out` zusätzlich auf stdout) hat `schema_version` 3. Er beginnt mit `schema_version`, `exit_code`, `complete`, `interrupted` und `scan_id`, gefolgt von Laufdaten (`scanner_version`, `signature_set`, `scan_path`, `results_dir`, `started_at`, `completed_at`, `duration_seconds`, `modes` und `counts`). `finding_counts` enthält `total`, `failing` (Funde auf oder über der Schwelle von `--fail-on`), `by_severity` und `by_type`. Alle Funde stehen in einer Liste, `findings`; jeder Eintrag hat eine `id`, einen `type` (`file`, `remote`, `directory`, `database` oder `blacklist`), eine `severity`, eine `confidence` und eine Liste `detections`. Eine Erkennung nennt die `signature_id` (`ISPP-…`), den `name`, die `engine` (`pattern`, `clamav`, `clamav-image`, `bayes`, `check` oder `hash`) und, soweit bekannt, die getroffenen Bereiche. Verworfene Erkennungen (False-Positive-Listen, `--ignore`, Übereinstimmung mit einer offiziellen Prüfsumme) werden in `suppressed` gezählt; `notes` erklärt Grenzen des Laufs, etwa eine nicht verfügbare Verifizierung oder eine gekürzte Liste. Die Ergebnisse der Wartungsprüfungen folgen in `software`, `wordpress_plugins`, `plugin_vulnerabilities` und `permission_issues`; `artifacts` listet die Dateien des Laufs auf und `errors` die fehlgeschlagenen Prüfungen. Treffer aus den Downloads der Tiefenprüfung haben den Typ `remote` mit `url` und `referenced_by`. Alle Schlüssel sind in snake_case. Ein Bericht ist höchstens 64 MiB groß: `errors` behält höchstens 1.000 Einträge, und ist der Bericht dann noch zu groß, werden zuerst die Wartungslisten und dann die Funde gekürzt, mit einem Hinweis `sections_truncated` oder `findings_truncated`.

**Scan-ID.** Eine wiederverwendete `--scan-id`, deren Bericht im Ergebnisverzeichnis bereits existiert, wird mit Exit-Code 2 abgelehnt.

**Statusdatei.** `--status-file` enthält eine einzelne Zeile, die atomar ersetzt wird: `PHASE;percent;hits;eta;scan_id`. Die Phasen sind `START`, `SCANNING`, `COMPLETE`, `INTERRUPTED` und `FAILED`; `COMPLETE` und `INTERRUPTED` erscheinen erst, wenn der JSON-Bericht existiert. `FAILED` bedeutet, dass der Scan ohne Bericht endete (der Exit-Code nennt den Grund); ein Scan, der vorzeitig stoppt, auch durch ein zweites Signal, hinterlässt dennoch `FAILED` oder `INTERRUPTED`. Die Datei hat den Modus `0644`.

**Fortschrittsstrom.** `--progress-fd` und `--progress-file` schreiben ein JSON-Objekt pro Zeile mit `event`, `scan_id` und `time`. Die Ereignisse sind `start` (mit `scanner_version`, `scan_path` und den geplanten `stages`), `stage_start`, `progress` (mit `stage` und `percent`, dazu `files_scanned`, `files_total`, `hits` und `eta_seconds` bei Dateien bzw. `database`, `database_index`, `database_total`, `table`, `table_index`, `table_total`, `rows` und `hits` bei Datenbanken), `stage_end` (mit `status` `ok`, `failed` oder `interrupted` sowie `error`) und `end` (mit `status`, `exit_code`, `findings` und `report`). Die Stages sind `files`, `permissions`, `timestamps`, `blacklist`, `software`, `plugins`, `plugin_vulnerabilities`, `database` und `report`. Die Stages `permissions`, `timestamps` und `blacklist` werden nur geplant, wenn die jeweilige Prüfung eingeschaltet ist. Fortschrittsereignisse kommen höchstens alle 250 ms pro Stage. Jeder Scan, der `start` ausgegeben hat, gibt auch `end` aus, einschließlich früher Fehlschläge. Ein Konsument, der hinterherhinkt, verliert `progress`-Ereignisse, niemals die anderen, und bremst den Scan nie aus. Der Deskriptor muss zum Schreiben geöffnet sein und wird nicht an Programme weitergereicht, die der Scanner startet. `--progress-file` weist einen Link, eine Pipe, ein Verzeichnis oder einen anderen nicht regulären Eintrag zurück, ersetzt eine vorhandene reguläre Datei durch eine neue mit Modus `0600` und weist einen Namen zurück, der auf `/` endet.

# 6. Ablauf eines Scans

1. **Dateisuche** — rekursive Traversierung der Scan-Wurzel mit Include-/Exclude-Filterung und einem persistenten Hash-Cache für inkrementelle Läufe.
2. **Scan nach binären Signaturen** — Prüfergebnisse des ClamAV-IPC-Dienstes.
3. **Mustererkennung** — 184 mitgelieferte Erkennungsregeln; reguläre Ausdrücke werden vor dem Scan kompiliert.
4. **CMS-Erkennung** — Identifikation von Plattform und Version.
5. **Plugin-Prüfung** — WordPress-Plugin-Versionen gegen eine Schwachstellendatenbank.
6. **Tiefenprüfung** — erweiterte Inhaltsanalyse (mit `--deep-check`).
7. **Bayes-Klassifikation** — ML-Erkennung (mit `--bayes-scan`).
8. **Vertrauenslisten** — SHA-256-basierte Reduzierung von False Positives: Eine Datei, die auf der offiziellen Liste bekannter Release-Dateien oder in Ihrer Whitelist steht, wird nicht gemeldet.
9. **Berichtswesen** — Ergebnisse werden zusammengestellt, zugestellt und archiviert.

**Besitzerprüfung.** Zusammen mit der Dateisuche prüft der Scanner, wem die Dateien gehören. Er meldet Dateien eines unerwarteten Kontos, das weder der Besitzer der Website noch root noch der Webserver-Benutzer ist (`ISPP-OWN-1`; `high` für Skriptdateien, sonst `medium`), Dateien des Webserver-Benutzers (`www-data`, `apache`, `httpd`, `nginx` oder `wwwrun`) unterhalb eines anders besessenen Verzeichnisses (`ISPP-OWN-2`), setuid- oder setgid-Bits an regulären Dateien (`ISPP-OWN-4`), Dateien, deren Besitzer kein Konto hat, an einer Stelle, an der kein Besitzer der Website bekannt ist, etwa direkt unterhalb eines Verzeichnisses von root (`ISPP-OWN-5`; unterhalb eines Website-Besitzers wird eine solche Datei als `ISPP-OWN-1` gemeldet), und Verzeichnisse mit 50 oder mehr leeren Skriptdateien (`ISPP-FS-1`). Ergebnisse für Dateien, die keine Skripte sind, werden pro Verzeichnis zusammengefasst. `--no-owner-check` schaltet die Prüfung ab; `--ignore` erfasst sie nicht.

**Berechtigungsprüfung.** Bei jedem Datei-Malware-Scan prüft der Scanner auch Berechtigungen, im lockeren Modus; `--check-permissions`, `--perm-check` und `--perm-check-strict` wählen den Modus und führen die Prüfung auch ohne Datei-Malware-Scan aus, `--no-check-permissions` schaltet sie ab. Gemeldet werden Konfigurationsdateien wie `wp-config.php`, auf die andere zugreifen können, sowie reguläre Dateien und Verzeichnisse unterhalb des Scan-Wurzelverzeichnisses, dieses eingeschlossen, die jeder beschreiben darf; symbolische Links werden übersprungen. Die Einträge für weltweit beschreibbare Objekte haben den Schweregrad `medium` und einen Vorschlag ohne das Schreibrecht für alle, aus `1777` wird etwa `1775`. Jede der drei Regeln listet höchstens 200 Einträge, die ersten in Pfadreihenfolge; gibt es mehr, enthält der Bericht einen Hinweis `permission_issues_truncated` mit der Zahl der ausgelassenen Einträge. `permissions_<scan-id>.txt` nennt die Gesamtzahl je Regel und die Einträge. Die Einträge stehen unter `permission_issues`, nicht unter den Funden: Sie ändern den Exit-Code nicht, und ihretwegen wird nichts in Quarantäne verschoben.

**Offizielle Prüfsummen.** Mit `--verify-checksums` vergleicht der Scanner WordPress-Core-Dateien und Plugin-Dateien mit den von wordpress.org veröffentlichten Prüfsummen. Er ruft `https://api.wordpress.org/core/checksums/1.0/` für den Core und `https://downloads.wordpress.org/plugin-checksums/<slug>/<version>.json` für Plugins ab; die Anfragen enthalten die WordPress-Version und die Sprache bzw. den Plugin-Slug und die Version sowie den User-Agent `ispprotect-malware-scan`. Die Listen werden zwischengespeichert, und ein Scan stellt höchstens 5.000 Anfragen, vier gleichzeitig. Eine Datei, die ihrer offiziellen Prüfsumme entspricht, wird nicht als Malware gemeldet; die verworfenen Erkennungen stehen in `suppressed` mit dem Grund `official_checksum_match`. Abweichungen sind Funde: eine veränderte Core-Datei (`ISPP-VFY-1`, `high`), eine Datei, die nicht zum Core gehört (`ISPP-VFY-2`, `high` für Skripte, sonst `low`), eine veränderte Plugin-Datei (`ISPP-VFY-3`, `medium`) und eine WordPress-Version, die sich nicht verifizieren lässt (`ISPP-VFY-4`, `low`). Kann eine Liste nicht abgerufen werden, weisen die Hinweise `verification_unavailable` oder `verification_unit_ignored` darauf hin. Die Option ist standardmäßig aus und wird für eine einzelne Datei mit einer Warnung ignoriert.

**Identische Kopien.** Eine Datei mit exakt demselben Inhalt wie eine Inhaltserkennung mit hoher Konfidenz wird mit der Erkennung `ISPP-HASH-1` gemeldet und wie das Original behandelt.

# 7. Erkennung von CMS, Plugins und Versionen

Der Versions-Scan erkennt die folgenden Plattformen und meldet veraltete Installationen samt ihrem genauen Speicherort — einschließlich vergessener Kopien in alten Ordnern:

WordPress · Joomla! · Drupal · Magento · MediaWiki · Contao · PrestaShop · Shopware · Roundcube · phpMyAdmin · WoltLab Board (wbb) · MySQLDumper · TYPO3 · CMS Made Simple

WordPress-Plugin-Versionen werden stets mit nebenläufigen Abfragen gegen eine Schwachstellendatenbank geprüft. Plugin-Berichte enthalten sowohl das Plugin-Verzeichnis als auch den Pfad der Hauptdatei, sodass Funde sofort bearbeitet werden können.

Einzelne Phasen deaktivieren Sie mit `--no-version-scan` (alle Versionsprüfungen) oder `--no-plugin-version-scan` (nur Plugin-Prüfungen).

# 8. Signatur-Feeds

Vor jedem Binär-Scan prüft der Scanner die Signatur-Feeds von Drittanbietern, die er auf dem Rechner vorhält, und lädt fehlende oder veraltete herunter. Jeder Feed liegt in einem eigenen Verzeichnis unter `/var/lib/malware-scan/sigs/`, zusammen mit dem Urhebervermerk des veröffentlichenden Projekts:

| Feed | Inhalt |
|---|---|
| Sanesecurity | Malware-Hashes sowie Makro-, URL- und Rogue-Datenbanken |
| PhishTank | Phishing-Datenbanken, abgeleitet aus PhishTank-Meldungen |

Downloads werden gegen die neben jeder Datenbank veröffentlichte Prüfsumme verifiziert, atomar geschrieben und vollständig übersprungen, solange ein Feed jünger als `--sig-update-interval` ist (standardmäßig 24 Stunden). Spiegelserver werden der Reihe nach probiert; ein fehlgeschlagener Feed blockiert nie einen Scan; ein nicht beschreibbares Verzeichnis (z. B. beim Betrieb als unprivilegierter Benutzer) wird übersprungen.

- `--no-sig-update` — Feed-Downloads vollständig abschalten
- `--update-signatures` — eine Aktualisierung erzwingen
- `--extended-signatures` — zusätzlich die Mail- und Spam-Datenbanken laden (erkennt mehr, aber mehr False Positives bei gehosteten Inhalten)
- `--signature-dir <dir>` — alternativer Speicherort

Signatur-Sets, die mit dem Produkt ausgeliefert statt heruntergeladen werden, sind unter `/var/lib/malware-scan/db/bundled/` installiert, ein Verzeichnis pro Projekt, jeweils mit seinem Lizenzhinweis. Zwei Sets von Drittanbietern sind mitgeliefert: das Projekt Yara-Rules (GPL-2.0) und die YARA-Regeln von ReversingLabs (MIT). Die Regeln sind für die Scan-Engine angepasst (Dateien zusammengeführt und umbenannt, von der Engine nicht auswertbare Regeln entfernt, Kommentare und Lizenzhinweise beibehalten). Die proprietären ISPProtect-Signaturen werden ebenfalls im Paket ausgeliefert, stammen aber von ISPProtect selbst.

# 9. Datenbank-Scan

Mit `--db-scan` untersucht der Scanner MySQL-/MariaDB-Inhalte auf eingeschleuste Malware und Spam-Links.

Konfiguration in `/etc/malware-scan/db-config.conf` (oder einer beliebigen Datei, die über `--db-scan-config` übergeben wird):

```
host = localhost
port = 3306
user = scanner
password = secret
databases = wordpress,joomla
```

Der Datenbankbenutzer benötigt Lesezugriff auf die aufgeführten Schemata. Die Kurzform `--db-scan=<file>` aktiviert den Scan mit genau dieser Konfigurationsdatei.

Die Datei verwendet die `my.cnf`-Syntax, einschließlich `[client]`-Abschnitten, Werten in Anführungszeichen und den Schlüsseln `socket` und `protocol`. Wie der `mysql`-Client erreicht der Scanner einen Server auf `localhost` über dessen Unix-Socket, sofern er einen findet (`/run/mysqld/mysqld.sock`, `/var/run/mysqld/mysqld.sock`, `/var/lib/mysql/mysql.sock`, `/run/mysql/mysql.sock`, `/tmp/mysql.sock`); `protocol = tcp` oder `host = 127.0.0.1` erzwingt TCP, ebenso ein anderer `port` als 3306, sofern kein `socket` konfiguriert ist (eine zweite Instanz auf eigenem Port).

Ein Passwort ist nicht nötig, wenn die `unix_socket`-Authentifizierung von MariaDB aktiviert ist, wie es für `root@localhost` unter Debian und Ubuntu der Fall ist:

```
malware-scan --path /var/www --db-scan --db-socket auto
```

Mit `--db-socket <path|auto>` wird die Konfigurationsdatei optional; als Benutzer gilt dann der konfigurierte, `--db-user` oder der Betriebssystembenutzer, der den Scan ausführt. Nennt `--db-user` einen anderen Benutzer als die Konfigurationsdatei, wird das Passwort aus dieser Datei nicht mitgeschickt.

**Was gemeldet wird.** Es gibt einen Fund je Datenbank, Tabelle, Spalte und Signatur, mit der Anzahl der treffenden Zeilen und höchstens drei Beispielzeilen. Eine Beispielzeile enthält den Schlüssel der Zeile (`pk`), einen SHA-256 der Zeile (`row_hash`), ein Stück des umgebenden Texts (`context`) und den getroffenen Text (`matched`). Ein Fund hat eine von zwei Klassen: `stored_payload` für Inhalte in gewöhnlichen Tabellen und `logged_attack` für Treffer in Log- und Statistiktabellen, die Anfragen aufzeichnen statt Inhalte, die die Website verwendet; diese werden mit dem Schweregrad `info` und ohne Beispielzeilen gemeldet. Arbeitstabellen von Sicherheits-Plugins, die eigene Signaturen und Scan-Ergebnisse speichern, werden nicht gelesen; Spalten mit Namen, die auf Zugangsdaten hindeuten, werden weder geprüft noch zitiert. Mit `--db-no-context` entfallen `context` und `matched`; Schlüssel und `row_hash` bleiben.

**Umfang.** `--db-scan-maxrows <n>` begrenzt die gelesenen Zeilen pro Tabelle (Standard 1000000, `0` liest alle Zeilen). `--db-only`, `--db-exclude`, `--db-only-table` und `--db-exclude-table` nehmen Muster entgegen und sind wiederholbar.

**Ausgabe.** Die Zusammenfassung im Terminal zeigt, wie viele Datenbanken, Tabellen und Zeilen untersucht wurden, welche Datenbanken und Tabellen übersprungen wurden und wie viele Erkennungen es gab. Die Einzelheiten stehen in `database_<scan-id>.txt` und erscheinen im Bericht als Funde vom Typ `database`. Die Datei ist tabulatorgetrennt mit den Spalten `Database`, `Table`, `Column`, `Signature`, `Name`, `Severity`, `Class`, `Rows` (alle treffenden Zeilen), `Key` (`column=value,...`), `RowHash` und `Context`, eine Zeile pro Beispielzeile. Datenbankfunde auf oder über der Schwelle von `--fail-on` zählen für Exit-Code 1 und sind Teil des E-Mail-Berichts.

# 10. E-Mail-Berichte

Scan-Berichte stellen Sie mit `--email-results <address>` per E-Mail zu (für mehrere Empfänger wiederholbar). Die SMTP-Einstellungen stehen in `/etc/malware-scan/email-config.conf` oder in einer beliebigen Datei, die über `--email-config` übergeben wird:

```
smtp_host = smtp.example.com
smtp_port = 587
smtp_user = alerts@example.com
smtp_password = secret
from_address = scanner@example.com
```

Der Scanner hat keinen eingebauten Scheduler: Er versendet die Ergebnisse am Ende eines Laufs, kombinieren Sie ihn also mit cron für tägliche, wöchentliche oder monatliche automatische Berichte.

### 10.1 DNS-Blacklists

`--blacklist-check` entnimmt die Domainnamen der Apache- und Nginx-Konfiguration und schlägt sie in DNS-Blacklists nach (Fund `ISPP-BL-1`, Schweregrad `medium`). Der Scanner bringt keine Blacklist-Zonen mit: Die kostenlosen öffentlichen Spiegel von Spamhaus und SURBL dürfen nicht kommerziell oder auf Shared-Hosting-Servern genutzt werden. Sie benötigen einen eigenen lizenzierten Zugang, etwa einen Schlüssel für den Spamhaus Data Query Service (DQS) oder einen SURBL-Datenfeed, den Ihr eigener Nameserver ausliefert, und tragen dessen Zonen in `/etc/malware-scan/blacklist.conf` oder in eine beliebige Datei ein, die über `--blacklist-config` übergeben wird. Das Paket installiert `blacklist.conf.example` als Vorlage:

```
[blacklist]
name = spamhaus-dbl
zone = <your-dqs-key>.dbl.dq.spamhaus.net
format = spamhaus-dbl

[blacklist]
name = surbl
zone = <your-surbl-zone>
format = surbl
```

Jeder `[blacklist]`-Eintrag hat drei Schlüssel:

- `name`: die Bezeichnung in Berichten, Ergebnisdateien und E-Mails; 1 bis 64 Buchstaben, Ziffern, `.`, `_` oder `-`, eindeutig innerhalb der Datei.
- `zone`: die DNS-Zone, in der die Domains nachgeschlagen werden. Eine DQS-Zone enthält Ihren Zugangsschlüssel.
- `format`: wie die Antworten gelesen werden, `spamhaus-dbl` für die Spamhaus DBL (auch über DQS) oder `surbl` für die SURBL-Multi-Liste.

Die Datei enthält höchstens 16 Einträge; die Zonen werden in der angegebenen Reihenfolge abgefragt. Da eine Zone einen Zugangsschlüssel enthalten kann, darf die Datei nur für ihren Besitzer lesbar sein (`chmod 600`) und muss root oder dem Konto gehören, unter dem der Scan läuft; andernfalls lehnt der Scanner sie ab. Die Zone wird nur für die DNS-Abfragen verwendet: Berichte, Ergebnisdateien, E-Mails und Meldungen zeigen den `name` des Eintrags, und der Fund nennt die Konfigurationsdatei in `config_file`. Die Suchreihenfolge ist dieselbe wie für die Datenbank- und E-Mail-Einstellungen (Abschnitt 11). Ohne verwendbare Konfiguration läuft die Prüfung nicht: Der Scan meldet den Fehler und endet mit Exit-Code 4, sofern es keine Funde gibt.

# 11. Konfigurationsassistent

Datenbank- und E-Mail-Einstellungen lassen sich interaktiv statt von Hand schreiben:

```
malware-scan --configure          # asks which part to configure
malware-scan --configure db       # database credentials only
malware-scan --configure email    # report delivery only
```

Der Assistent fragt jeden Wert ab (und bietet den aktuell konfigurierten Wert als Vorgabe an) und **prüft die Einstellungen anschließend am echten Server**: Er verbindet sich mit der Datenbank und listet die Schemata auf, öffnet die SMTP-Sitzung und authentifiziert sich, ohne etwas zu versenden. Eine fehlgeschlagene Prüfung wird gemeldet und lässt sich übergehen, sodass ein Host vorbereitet werden kann, bevor der Server existiert, mit dem er spricht.

Die Dateien werden mit dem Modus `0600` geschrieben — sie enthalten Passwörter —, und zwar nach `/etc/malware-scan/`, wenn dieses beschreibbar ist, andernfalls in das Konfigurationsverzeichnis des Benutzers (`~/.config/malware-scan/`). Mit `--configure-dir` legen Sie sie an einem anderen Ort ab.

Sind sie einmal geschrieben, übernehmen Scans beide Dateien automatisch; der Scan gibt aus, welche Datei er verwendet hat. Die Suchreihenfolge lautet:

1. der auf der Kommandozeile angegebene Pfad (`--db-scan-config`, `--email-config`, `--blacklist-config`)
2. das Konfigurationsverzeichnis dieses Kontos
3. `/etc/malware-scan/`
4. unter `sudo` das Konfigurationsverzeichnis des aufrufenden Kontos
5. nur für Datenbanken die MySQL-Standardorte des Systems (`/etc/mysql/debian.cnf`, `/root/.my.cnf`, `~/.my.cnf`)

Eine Datei, die existiert, aber einem anderen Konto gehört, wird übersprungen, statt den Scan scheitern zu lassen. Wird nichts gefunden, listet der Fehler jeden versuchten Pfad auf.

# 12. Performance und Tuning

- **Nebenläufigkeit** — `--concurrency <n>` steuert parallele Datei-Scans (Standard 5, max. 1024). Höhere Werte beschleunigen große Verzeichnisbäume auf Kosten der E/A-Last.
- **Inkrementelle Scans** — `--incremental` verwendet den persistenten Clean-/Malware-Hash-Zustand wieder, sodass unveränderte Dateien nicht erneut untersucht werden.
- **Temporäre Dateien** — `--use-tmpfs` bevorzugt `/dev/shm`; `--use-tmp <dir>` nennt ein alternatives Verzeichnis.
- **Filterung** — `--exclude` / `--include` akzeptieren Globs mit `*` und `**` und sind wiederholbar; `--exclude-from` / `--include-from` lesen Muster aus Dateien.
- **Laufzeit** — `--timing` gibt die Gesamtlaufzeit des Scans für Benchmarks aus.

# 13. Ergebnisse, Quarantäne und Whitelist

**Berichte.** Jeder Scan schreibt `scan_report_<scan-id>.json` sowie Artefakte pro Phase (einschließlich leerer Berichte) standardmäßig unter `/var/log/malware-scan`. Den Speicherort ändern Sie mit `--results-dir` oder `MALWARE_SCAN_RESULTS_DIR`; `--json-file` kopiert den vollständigen Bericht zusätzlich an einen beliebigen Ort. `--quiet` unterdrückt nur Terminalausgaben, niemals Berichte. `--cleanup [days]` löscht Berichte, die älter als diese Anzahl Tage sind (ohne Wert 7); das Ergebnisverzeichnis wird dabei nie angelegt.

**Ergebnisdateien.** Jeder Dateiname enthält die Scan-ID. Je nach durchgeführten Prüfungen enthält das Ergebnisverzeichnis `found_malware_<scan-id>.txt`, `database_<scan-id>.txt`, `permissions_<scan-id>.txt`, `timestamps_<scan-id>.txt`, `blacklists_<scan-id>.txt`, `software-versions_<scan-id>.txt`, `wordpress-plugins_<scan-id>.txt`, `plugin-vulnerabilities_<scan-id>.json` und, wenn Dateien in Quarantäne verschoben wurden, `quarantine-manifest-<scan-id>.json`. Das Verzeichnis hat den Modus `0700`, die Dateien den Modus `0600`. Textdateien sind tabulatorgetrennt, beginnen mit einer Kopfzeile und schreiben Steuerzeichen in maskierter Form. `found_malware_<scan-id>.txt` hat eine Zeile je Datei mit einer Erkennung auf oder über der Schwelle von `--fail-on` und die Spalten `FilePath`, `Signature`, `Name`, `Severity`, `DetectedAt` (RFC 3339) und `Scanner` (`binary`, `binary-image`, `pattern`, `bayes`, `check` oder `hash`). Die Spalten von `database_<scan-id>.txt` stehen in Abschnitt 9.

**Quarantäne.** `--quarantine <dir>` (oder `--quarantine-mode`) verschiebt erkannte lokale Dateien aus dem Webroot; `--restore` stellt sie zusammen mit `--quarantine` wieder her. Verschoben werden nur Dateien mit einer Erkennung auf oder über der Schwelle von `--fail-on`, und nur bei einer Muster-, Binärsignatur- oder Bayes-Erkennung, bei den Inhaltsprüfungen `ISPP-CHK-3` und `ISPP-CHK-4` und bei einer identischen Kopie (`ISPP-HASH-1`). Alles andere wird nur gemeldet und nie verschoben: Besitzer- und Dateisystem-Funde (`ISPP-OWN-*`, `ISPP-FS-*`), Prüfsummen-Ergebnisse (`ISPP-VFY-*`), die Datei- und Ortsprüfungen `ISPP-CHK-1` und `ISPP-CHK-2`, Zeitstempel- und Blacklist-Funde, Datenbankfunde sowie die Ergebnisse der Versions-, Plugin- und Berechtigungsprüfungen. Eine Datei, die sich nach dem Scan geändert hat oder größer als 512 MiB ist, bleibt an ihrem Platz. Eine Datei gilt nur dann als in Quarantäne, wenn das Original verschwunden ist; hat sie weitere Hardlinks, wird eine Warnung ausgegeben. Ein Quarantäne-Verzeichnis, das den Scan-Pfad enthält, ist ein Konfigurationsfehler (Exit 2). Quarantäne- und Ergebnisverzeichnis werden vom Scan ausgenommen, und ein unterbrochener Scan verschiebt nichts. Der E-Mail-Bericht listet die Dateien mit Erkennungen auf oder über der Schwelle von `--fail-on`.

**Signaturen ignorieren.** `--ignore <name>` nimmt eine Signatur-ID (zum Beispiel `ISPP-…`) oder einen Signaturnamen entgegen und ist wiederholbar. Es unterdrückt Muster-, Binärsignatur- und Bayes-Erkennungen, die Prüfungen `ISPP-CHK-1` bis `ISPP-CHK-4` und Datenbankfunde dieser Signatur. Es erfasst weder Besitzer- (`ISPP-OWN-*`, `ISPP-FS-*`) noch Prüfsummen-Funde (`ISPP-VFY-*`), weder Zeitstempel- noch Blacklist-Funde und auch nicht die Ergebnisse der Berechtigungs-, Versions- und Plugin-Prüfungen. Unterdrückte Erkennungen werden in `suppressed` gezählt.

**Whitelist.** Verifiziert saubere Dateien können von künftigen Treffern ausgenommen werden: `--whitelist <file>` fügt den SHA-256 einer Datei zu `~/.ispp_scan.whitelist` (oder der mit `--whitelist-path` genannten Datei) hinzu und beendet sich. Die Whitelist hält legitime Dateien über mehrere Läufe hinweg ruhig.

# 14. Proben einreichen

Proben nicht erkannter Malware und von False Positives helfen, die Erkennung zu verbessern:

```
malware-scan --report <file> --confirm-file-upload
malware-scan --false-positive <file> --confirm-file-upload --email you@example.com
```

Beide senden die vollständige ausgewählte Datei an ISPProtect und erfordern daher das ausdrückliche Flag `--confirm-file-upload`. Dateien größer als 10.048.000 Bytes sowie Symlinks werden abgelehnt, bevor eine Anfrage gestellt wird.

# 15. ISPProtect Console (malware-agent)

## 15.1 Was es ist

`malware-agent` ist ein optionales, eigenständiges Paket, das einen Server mit der **ISPProtect Console** unter `https://license.ispprotect.com` verbindet. Die Console zeigt Scan-Berichte verknüpfter Server an einer Stelle und kann auf Anforderung Scans starten, Funde prüfen und Reparaturen ausführen. Der Scanner arbeitet auch ohne den Agenten vollständig; die Installation des Scanners installiert oder aktiviert ihn nicht.

Der Agent baut ausschließlich ausgehende HTTPS-Verbindungen auf. Er öffnet keinen TCP- oder UDP-Listener und fragt die Console regelmäßig nach Aufträgen.

Voraussetzungen:

- Linux mit systemd und Root-Rechten (der Produktiv-Agent verweigert den Start unter einem anderen Benutzer).
- Ein lokales Dateisystem mit Unterstützung für Linux-Datei-Leases. Für Reparaturen und Wiederherstellungen muss `fs.lease-break-time` mindestens 5 Sekunden betragen; andernfalls verweigert der Agent sichere Reparaturen.
- Ein installiertes Paket `malware-scan`, außerdem `ca-certificates`, `systemd` und PHP (`php-cli`, unter EL `/usr/bin/php`). Node.js wird empfohlen. Der Paketmanager installiert diese als Abhängigkeiten; PHP und Node.js prüfen die Syntax von Reparaturkandidaten, und ohne den Interpreter werden Reparaturen dieses Dateityps abgelehnt.
- Eine aktive kostenpflichtige Lizenz: Jahreslizenz oder Scan-Pakete. Testlizenzen lassen sich nicht verknüpfen. Abgelaufene oder gesperrte Lizenzen behalten den Zugriff auf Berichte und lokale Wiederherstellungen; neue Scans und KI-Anfragen erfordern eine aktive kostenpflichtige Lizenz.

## 15.2 Installation

Installieren Sie das Paket aus demselben Repository wie den Scanner (Abschnitt 2):

```
apt-get install malware-agent       # Debian / Ubuntu
dnf install malware-agent           # EL8 / EL9 / EL10
```

Das Paket registriert den Server nicht und startet den Dienst nicht; beides sind ausdrückliche Schritte des Administrators.

## 15.3 Einen Server verknüpfen

Die Erstregistrierung übergibt die Adresse der Console, Ihre Lizenzdatei des Scanners und die Website-Verzeichnisse, in denen die Console arbeiten darf:

```
sudo malware-agent --link \
  --portal https://license.ispprotect.com \
  --license-file /etc/malware-scan/license.key \
  --root /var/www
sudo systemctl enable --now malware-agent
```

Verwenden Sie die tatsächliche Lizenzdatei des Scanners; der Schlüssel selbst erscheint nie auf der Kommandozeile. Wiederholen Sie `--root` für jedes Website-Verzeichnis (zwischen 1 und 32). Roots müssen vorhandene, absolute Verzeichnisse ohne Symlinks sein; `/` und das Zustandsverzeichnis des Agenten werden abgelehnt. Die Scanner-Programmdatei ist standardmäßig `/usr/bin/malware-scan` und lässt sich mit `--scanner` ändern.

Der Befehl gibt eine URL und einen Einmal-Code aus. Öffnen Sie die URL oder geben Sie den Code in der Console ein, und bestätigen Sie, dass der angezeigte Hostname Ihrem Server entspricht.

Für jede spätere Anmeldung oder um ein weiteres Konto hinzuzufügen, führen Sie aus:

```
sudo malware-agent --link
```

Codes laufen nach zehn Minuten ab und sind nur einmal verwendbar. **Ein Code gewährt die volle Kontrolle über diesen Server; halten Sie ihn daher geheim.** Ein Konto kann mehrere Server sammeln, aber jeder Server benötigt seinen eigenen Code, der lokal auf diesem Server erzeugt wird.

Mit einem neuen Code lässt sich ein Server in ein anderes Konto verschieben. Das bisherige Konto verliert den Zugriff, und seine ausstehenden Aufträge und Zeitpläne werden abgebrochen. Der neue Eigentümer erbt die gesamte Server-Historie, einschließlich Fallnachrichten und aufbewahrter Proben. Die Einwilligung zum Datei-Upload und automatische Richtlinien werden bei der Übertragung zurückgesetzt.

### 15.4 Was die Console auf dem Server tun kann

Der Agent akzeptiert nur typisierte Aufträge; das Protokoll kennt keine Möglichkeit, Shell-Befehle zu senden.

| Auftrag | Wirkung |
|---|---|
| Scan | Führt den installierten Scanner auf einem registrierten Root ohne Interaktion aus. Datenbank-Scans nutzen die lokale Datenbankkonfiguration des Scanners; die Console kann weder Zugangsdaten noch SQL vorgeben. |
| Datei-Erfassung | Liest eine ausgewählte Datei und sendet deren Inhalt an die Console. Erfordert eine Einwilligung (Abschnitt 15.5). |
| Reparatur | Ersetzt eine geprüfte Datei, nachdem die Console den vollständigen Vorher-/Nachher-Text gezeigt hat. Zuvor wird ein nur für Root lesbares Backup angelegt. |
| Wiederherstellung | Spielt eine gesicherte Datei zurück. Eine neuere Version der Datei wird nie überschrieben. |
| Ignorieren | Fügt einen hash-spezifischen (`sha256:`) Whitelist-Eintrag hinzu, der nur Scans betrifft, die der Agent startet. |

Einschränkungen: Der Agent fasst nur Dateien unterhalb der registrierten Roots an und liest sie auf den Root beschränkt. Symlinks, die aus einem Root hinausführen, folgt er nicht, und Spezialdateien lehnt er ab. Eine Reparatur setzt eine unveränderte Datei (gleiches SHA-256) voraus, eine reguläre Datei von höchstens 512 KiB ohne Hardlinks; binäre und leere Dateien sowie Dateien mit setuid, setgid oder Capabilities werden abgelehnt. Während des kurzen Schreibvorgangs ist das enthaltene Verzeichnis kurzzeitig auf Root beschränkt, sodass der Zugriff auf Dateien darin für einen Moment unterbrochen sein kann. Belegte Dateien, nicht unterstützte Dateisysteme und geänderte Dateien führen zu einem sicheren Abbruch: Es wird nichts geändert.

### 15.5 Datenschutz: Was gesendet wird

- **Bei der Registrierung:** der Lizenzschlüssel (aus der Lizenzdatei gelesen), der Hostname, die konfigurierten Roots, die Agent-Version und ein Geräte-Token, das der Agent lokal erzeugt hat und später zur Authentifizierung nutzt. Die Agent-Version wird bei jeder Abfrage gesendet.

- **Berichte und Funde:** Ergebnisse von Scans, die aus der Console gestartet wurden, und fertige Berichte `scan_report_<scan-id>.json` des Scanners (Abschnitt 15.6) enthalten Metadaten und Funde, etwa Dateipfade, Hashes, Signaturnamen und Software-Versionen. Dateiinhalte enthalten sie nicht. Datenbank-Ausschnitte und lokale Artefaktpfade werden aus importierten Berichten entfernt, und Portal-Scans lassen die Treffer-Inhalte der Datenbank weg. Ein Bericht enthält höchstens die 10.000 wichtigsten Funde; ein Hinweis nennt die Zahl der ausgelassenen. Während ein Scan läuft, werden Fortschrittsereignisse gesendet.
- **Dateiinhalte:** nur nachdem ein berechtigter Console-Benutzer die Upload-Einwilligung erteilt hat. Bis dahin ist der Upload deaktiviert. Ausgewählte Dateien können privaten Code oder Zugangsdaten enthalten. Die Console speichert Proben verschlüsselt und bewahrt sie für ihre eigene Aufbewahrungsdauer auf. Ein Widerruf der Einwilligung bricht ausstehende Upload-, KI- und Reparaturaufträge ab.
- **KI-Funktionen:** nur für Dateien, für die Sie zugestimmt haben. KI-Aufträge mit Einwilligung gehen an OpenRouter und an die in der Console konfigurierten Modellanbieter. KI-Bewertungen und Reparaturvorschläge verbrauchen Punkte, die sich Lizenz-weit teilen. Ein Vorschlag ist kein Beleg dafür, dass eine Website sicher ist.
- **Wer den Server kontrolliert:** Wer einen gültigen Verknüpfungscode besitzt oder Zugriff auf das verknüpfte Console-Konto hat, kontrolliert den Server im Rahmen von Abschnitt 15.4. Gehen Sie mit Codes und Konten entsprechend um. Die Upload-Einwilligung kann ein Console-Benutzer erteilen, der mit einem Verknüpfungscode dieses Servers angemeldet ist.

## 15.6 Einstellungen und Dateien

| Element | Standard |
|---|---|
| Konfiguration | `/etc/malware-agent/config.json`, Modus `0600`, Eigentümer root |
| Zustand, Backups, Wiederherstellungsjournale | `/var/lib/malware-agent` (Modus `0700`) |
| Verzeichnisse für importierte Berichte | `/var/log/malware-scan` |
| Lokale Backup-Aufbewahrung | 30 Tage |

Die Datei enthält ein geheimes Geräte-Token; geben Sie sie nicht weiter und fügen Sie ihren Inhalt nie in ein Ticket ein. Schlüssel, die Sie ändern dürfen:

| Schlüssel | Bedeutung |
|---|---|
| `report_dirs` | Vertrauenswürdige Verzeichnisse, aus denen Berichte des Scanners (`scan_report_<scan-id>.json`) automatisch importiert werden. Wird bei der Erstregistrierung mit wiederholtem `--report-dir` gesetzt; `"report_dirs": []` deaktiviert den Berichtsimport. Die Verzeichnisse müssen außerhalb der Website-Roots liegen. Berichte mit anderer Schema-Version werden nicht importiert; führen Sie einen neuen Scan mit dem aktuellen Scanner aus. |
| `backup_days` | Aufbewahrungsdauer der Backups in Tagen, 1 bis 365. Für die Wiederherstellung benötigte Backups bleiben länger erhalten. |
| `roots` | Die registrierten Website-Roots. Die Konfiguration speichert zusätzlich die Identität (Gerät und Inode) jedes Roots, ändern Sie diesen Schlüssel daher nicht von Hand. Um Roots zu ändern, stoppen Sie den Dienst und erstellen mit dem Registrierungsbefehl eine neue Konfiguration; bewahren Sie die alte Konfiguration und die Backups auf, bis eine Wiederherstellung nicht mehr nötig ist. |

Mit `--state-dir` und `--config` wählen Sie bei der Registrierung andere Speicherorte. Starten Sie den Dienst nach einer Änderung der Konfiguration neu.

### 15.7 Einen Server trennen

1. Verwenden Sie in der Console **Disconnect server**. Dadurch wird das Geräte-Credential des Servers widerrufen.
2. Führen Sie auf dem Server `sudo systemctl disable --now malware-agent` aus.

Einen lokalen Befehl zum Trennen gibt es nicht. Um den Server erneut zu verbinden, erstellen Sie mit dem Registrierungsbefehl aus Abschnitt 15.3 eine neue private Konfiguration. Das Entfernen des Pakets stoppt den Dienst, behält aber Konfiguration, Backups und Wiederherstellungsjournale.

### 15.8 Umstieg von malware-daemon

`malware-agent` ersetzt das frühere Paket `malware-daemon`; beide lassen sich nicht gemeinsam installieren. Stoppen und deaktivieren Sie zuerst den alten Dienst:

```
sudo systemctl disable --now malware-daemon.service
# Paket malware-agent mit Ihrem Paketmanager installieren
sudo systemctl enable --now malware-agent.service
```

Ohne `--config` verwendet der Agent `/etc/malware-agent/config.json` und andernfalls weiterhin `/etc/malware-daemon/config.json`. Eine ungültige Konfiguration ist ein Fehler und löst keinen Fallback aus. Bestehende Registrierung, Zustandsverzeichnis, Backups und Wiederherstellungsjournale bleiben an ihrem Platz; verschieben oder löschen Sie sie nicht. Neue Registrierungen verwenden `/etc/malware-agent/config.json` und `/var/lib/malware-agent`.

### 15.9 Dienstbetrieb

```
systemctl status malware-agent
journalctl -u malware-agent --since today
```

`malware-agent --version` gibt die Agent-Version aus. Meldet die Wiederherstellung einen geänderten Inode, stoppen Sie den Dienst und bewahren Sie `directory-guard.json`, `file-metadata.json` und `backups/` im Zustandsverzeichnis zur Untersuchung auf; löschen Sie kein Journal, um den Fehler zum Schweigen zu bringen.

# 16. Daten, Zustand und Umgebung

| Zweck | Standard | Überschreiben |
|---|---|---|
| Scan-Berichte | `/var/log/malware-scan` | `--results-dir`, `MALWARE_SCAN_RESULTS_DIR` |
| Signatur-Feeds | `/var/lib/malware-scan/sigs/` | `--signature-dir` |
| Mitgelieferte Signatur-Sets | `/var/lib/malware-scan/db/bundled/` | — |
| Datendateien (Signatur-DB, Whitelist, Klassifikator) | `/var/lib/malware-scan`, `/usr/share/malware-scan`, `/etc/malware-scan`, danach relativ zum Binary und zum Arbeitsverzeichnis | `MALWARE_SCAN_DATA_DIR` (durch Doppelpunkt getrennt) |
| Beschreibbarer Zustand (z. B. `classifier.model.user`) | `MALWARE_SCAN_STATE_DIR`, dann `/var/lib/malware-scan`, XDG-Orte, Konfigurationsverzeichnis des Benutzers, Arbeitsverzeichnis | `MALWARE_SCAN_STATE_DIR` |
| Zustand inkrementeller Scans (`--incremental`) | `cache/<sha1 of the absolute scan root>.state` im ersten beschreibbaren Zustandsverzeichnis von oben | `MALWARE_SCAN_STATE_DIR` |
| Lizenzschlüssel | gespeicherter Schlüssel | `MALWARE_SCAN_LICENSE_KEY`, `MALWARE_SCAN_LICENSE_FILE` |
| Auffinden des Verdict-Dienstes | neben dem Scanner-Binary, `/usr/libexec/malware-scan`, `/usr/lib/malware-scan`, `PATH` | `MALWARE_SCAN_CLAMAV_IPC` (expliziter Pfad) |
| Agent-Konfiguration (`malware-agent`) | `/etc/malware-agent/config.json` (Modus `0600`) | `--config` bei der Registrierung |
| Agent-Zustand, Backups, Wiederherstellungsjournale | `/var/lib/malware-agent` | `--state-dir` bei der Registrierung |
| Berichtsimport des Agenten | `/var/log/malware-scan` | `report_dirs` in der Agent-Konfiguration (Abschnitt 15.6) |
| Externer Verdict-Dienst | Kindprozess wird pro Scan gestartet | `MALWARE_SCAN_CLAMAV_IPC_SOCKET` (einen bereits laufenden Dienst verwenden) |

Paketinstallationen liefern Signaturen in `/var/lib/malware-scan/` aus; portable Installationen verwenden weiterhin die lokalen Ordner `db/` und `data/`. Umgebungsvariablen überschreiben all das — nützlich, wenn mehrere Scanner-Instanzen mit unterschiedlichen Signatur-Sets laufen.

Der inkrementelle Zustand wird anhand des absoluten, bereinigten Scan-Pfads zugeordnet, sodass `--path .`, `--path /var/www/site/` und `--path /var/www/site` eine gemeinsame Zustandsdatei nutzen.

## 16.1 Betrieb in einer Sandbox

Unter systemd-Härtung, in einem Container oder mit einem seccomp-/AppArmor-Profil benötigt der Scanner:

- **Lesezugriff** auf die Scan-Wurzel sowie auf `/var/lib/malware-scan`, `/usr/share/malware-scan` und `/etc/malware-scan`.

- **Schreibzugriff** auf das Ergebnisverzeichnis (`/var/log/malware-scan`), das Zustandsverzeichnis (bei `--incremental`, Training oder Feed-Updates; richten Sie `MALWARE_SCAN_STATE_DIR` auf einen beschreibbaren Pfad), das Quarantäne-Verzeichnis, falls verwendet, und ein temporäres Verzeichnis (`TMPDIR`, `--use-tmp` oder `--use-tmpfs`).
- **Den Verdict-Dienst.** Der Scanner startet `clamav-ipc-service` als Kindprozess, daher müssen `fork` / `exec` erlaubt und das Binary in `/usr/libexec/malware-scan` oder `/usr/lib/malware-scan` ausführbar sein. Der Dienst liest die Signaturverzeichnisse (`/var/lib/malware-scan/db`, `/var/lib/malware-scan/sigs`) und jede Datei, nach der er gefragt wird, und lauscht auf einem Unix-Socket in einem privaten Verzeichnis unter `TMPDIR`, `XDG_RUNTIME_DIR`, `/tmp` oder `/run` (das erste, das funktioniert; der Pfad muss unter 100 Bytes bleiben). Scanner und Dienst müssen dasselbe temporäre Verzeichnis sehen. Der Dienst wird beendet, wenn der Scanner endet. Alternativ betreiben Sie einen Dienst außerhalb der Sandbox und setzen `MALWARE_SCAN_CLAMAV_IPC_SOCKET` auf dessen Socket.
- **Netzwerk (HTTPS):** immer `license.ispprotect.com` und `license2.ispprotect.com`; `api.wordpress.org` für Plugin-Versionen und, mit `--verify-checksums`, die Core-Prüfsummen sowie `downloads.wordpress.org` für die Plugin-Prüfsummen; die Spiegelserver der Signatur-Feeds, außer bei `--no-sig-update`; und mit `--deep-check` die in gescannten Dateien gefundenen URLs, außer bei `--no-deep-check-network`. `--blacklist-check` benötigt nur DNS: Die Domains werden über den System-Resolver in den Zonen aus `blacklist.conf` nachgeschlagen. Datenbank-Scans benötigen den Datenbank-Socket oder den TCP-Port.
- **Netzwerk für `malware-agent` (Abschnitt 15):** ausgehend HTTPS (Port 443) zur ISPProtect Console unter `license.ispprotect.com`; der Agent nimmt selbst keine Verbindungen an.

# 17. Der ClamAV-Verdict-Dienst

`clamav-ipc-service` ist ein eigenständiges, GPL-lizenziertes Programm, das `libclamav` einbindet. Die Debian-Pakete installieren es als Abhängigkeit von `malware-scan`. Es lässt sich auch eigenständig installieren und verwenden:

```
apt-get install clamav-ipc-service
```

Fehlt er, meldet der Scanner, dass die Binär-Engine nicht verfügbar ist, und setzt den Scan mit Muster-, Bayes- und Datenbank-Scan fort. Der Dienst wird pro Scan gestartet und danach beendet; richten Sie `MALWARE_SCAN_CLAMAV_IPC_SOCKET` auf eine laufende Instanz, um einen Dienst über mehrere Scans hinweg gemeinsam zu nutzen.

# 18. Fehlerbehebung

**„Binary engine unavailable“** — `clamav-ipc-service` ist nicht installiert oder nicht auffindbar. Installieren Sie das Paket oder setzen Sie `MALWARE_SCAN_CLAMAV_IPC`. Scans laufen mit den übrigen Engines weiter.

**Konfigurationsdatei nicht gefunden** — führen Sie `malware-scan --configure` aus oder prüfen Sie die Suchreihenfolge in Abschnitt 11; der Fehler listet jeden versuchten Ort auf.

**Zu viele False Positives bei gehosteten Mail-/Spam-Inhalten** — vermeiden Sie `--extended-signatures` und setzen Sie verifiziert saubere Dateien auf die Whitelist (Abschnitt 13).

**Scan einer einzelnen Datei schlägt fehl** — Scans einzelner Dateien erfordern `--allow-single`.

**Unerklärliche Last zwischen Scans** — Feeds werden höchstens alle `--sig-update-interval` Stunden aktualisiert; verwenden Sie `--no-sig-update` auf Hosts mit begrenzter Bandbreite. Ziehen Sie für große Verzeichnisbäume `--incremental` und `--use-tmpfs` in Betracht.

**Berechtigungsprobleme als unprivilegierter Benutzer** — nicht beschreibbare Feed-Verzeichnisse werden automatisch übersprungen; Laufberichte landen in einem für den Benutzer beschreibbaren `--results-dir`.

**Agent: „enrollment is incomplete; run malware-agent --link --config ...“** — der Server wurde nie verknüpft. Führen Sie `sudo malware-agent --link` wie in Abschnitt 15.3 beschrieben aus und starten Sie danach den Dienst.

**Agent: „agent configuration must be a regular file readable only by its owner (0600)“** — korrigieren Sie die Berechtigungen: `chmod 600 /etc/malware-agent/config.json`, Eigentümer root.

**Agent: „portal URL must use HTTPS“** — übergeben Sie `--portal https://license.ispprotect.com`; es werden nur HTTPS-Adressen ohne Pfad oder Query akzeptiert.

**Agent: „production agent must run as root“** — führen Sie den Befehl mit `sudo` aus und betreiben Sie den Dienst als root.

**Agent: „managed root identity changed: <path>; update the local configuration after verifying the directory“** — das Verzeichnis an diesem Pfad ist nicht das registrierte (zum Beispiel nach einer Speichermigration). Prüfen Sie es, stoppen Sie dann den Dienst und erstellen Sie mit dem Registrierungsbefehl eine neue Konfiguration (Abschnitt 15.6).

**Agent: „portal returned <status>: <message>“** — die Console hat die Anfrage abgelehnt, etwa weil die Lizenz eine Testlizenz oder nicht aktiv ist oder das Credential widerrufen wurde. Ein abgelaufener (zehn Minuten) oder bereits verwendeter Verknüpfungscode wird ebenfalls abgelehnt: Führen Sie `sudo malware-agent --link` erneut aus, um einen neuen Code zu erhalten.

**Agent: „safe repairs require fs.lease-break-time of at least 5 seconds“** — erhöhen Sie die Kernel-Einstellung `fs.lease-break-time` auf 5 oder mehr.

# 19. Support und Rechtliches

- Die Dokumentation wird nach `/usr/share/malware-scan/` installiert
- Support: support@ispprotect.com · ispprotect.com
- Anbieterangaben: Imprint / Impressum
- Proben von Malware und False Positives: `--report` / `--false-positive` (Abschnitt 14)

ISPProtect ist proprietäre Software der Projektfarm GmbH. Signatur-Sets von Drittanbietern bleiben unter ihren eigenen Lizenzen und Urhebervermerken (Sanesecurity, PhishTank, Yara-Rules, ReversingLabs); `clamav-ipc-service` wird unter der GPL in einem eigenen Paket und Repository verteilt.