Cerebion Rivet — Quick Reference Card

Before You Scan

Always work on a **copy or dedicated branch** — never apply AI-generated fixes directly to production systems or your main branch. Review all fixes before committing.

Risk Score Guide

Score	Level	Action
0–14	Minimal	Quantum-resistant or very low risk
15–39	Low	Monitor; no immediate action
40–59	Medium	Plan migration within 3–5 years
60–79	High	Prioritize migration now
80–100	Critical	Begin migration immediately

Score = Algorithm Risk (40%) + Timeline Risk (25%) + Business Impact (20%) + PQC Readiness (15%)

Algorithm Risk at a Glance

Algorithm	Score	Status
RSA-1024	95	Critically vulnerable — breakable by 2026–2028
RSA-2048	88	Highly vulnerable — breakable by 2032–2035
RSA-4096	78	Still fully broken by Shor's — needs more qubits
ECDSA P-256 / ECDH	90	Fewer qubits needed than RSA — equally critical
ECDSA P-384 / P-521	80 / 72	Larger curve, slightly longer timeline
DSA / DH	85–95	Fully broken by Shor's algorithm

Algorithm	Score	Status
SHA-256	50	Grover reduces the effective strength
AES-128	30	Grover halves security to 64-bit
MD5 / SHA-1	100	Classically and quantum broken
AES-256 / SHA-384+	15–25	Quantum-resistant; small residual score

---

Post-Quantum Replacements (NIST FIPS)

Replace	With	Standard
RSA-OAEP / ECDH / DH	**ML-KEM** (Kyber)	FIPS 203
RSA / ECDSA / DSA	**ML-DSA** (Dilithium)	FIPS 204
ECDSA (stateless)	**SLH-DSA** (SPHINCS+)	FIPS 205
AES-128	**AES-256** / ChaCha20-Poly1305	—
SHA-1 / MD5	**SHA-384** / SHA-512 / SHAKE-256	—

Quantum Threat Timeline

Year	Threat Level	Assumed quantum computer
2024	Minimal	~100 logical qubits
2026	Low	~1,000 logical qubits
2030	Medium	~4,000 logical qubits
2033	High	~10,000 logical qubits
2040	Critical	~100,000 logical qubits

The scoring uses these estimates; they are planning assumptions, not predictions.

Supported Binary Formats

PE (.exe, .dll) · ELF · Mach-O (.dylib, .app) · .NET · Java .class / JAR · DEX (Android)

Supported Code Languages

46 languages and configuration formats, including: Python · JavaScript · TypeScript · Java · C/C++ · C# · Go · Rust · Ruby · PHP · Kotlin · Scala · Swift · Bash · Lua · Perl

· PowerShell · R · VB · COBOL · Groovy · Apex · Elixir · Clojure · OCaml · Scheme · Vue · Dart · Solidity · Cairo · Circom · Move · Terraform · HCL · Dockerfile · SQL · JSON · YAML · TOML · XML · Jsonnet · .env · Protocol Buffers

---

Key Management Analyzer — Supported Backends

HashiCorp Vault · AWS KMS · AWS Certificate Manager · Azure Key Vault · Azure Managed HSM · GCP Cloud KMS · Kubernetes Certificate Manager (cert-manager) · Microsoft AD CS

Requires a read-only, customer-issued credential per backend. Credentials are stored only in your OS credential manager (never in Rivet's database, never sent to Cerebion), and the connector cannot decrypt, sign, or export — read-only by design. On-premise only.

Cloud Analyzer — Supported Providers

Provider	Services
AWS	ELB · CloudFront · API Gateway · S3 · RDS · EBS · Secrets Manager · CloudHSM
Azure	Application Gateway · Storage Account · SQL Server
GCP	Cloud Load Balancing · Cloud Storage · Cloud SQL · Secret Manager

AWS ACM and KMS are scanned by the Key Management Analyzer.

Requires a read-only, customer-issued credential (AWS IAM key, Azure Service Principal, or GCP Service Account). Credentials stored only in your OS credential manager — never in Rivet's database, never sent to Cerebion.

Binary Analysis Confidence Levels

Confidence	Meaning
High	YARA + binary analysis both succeeded
Medium	One method succeeded
YARA Primary	YARA matched; binary analysis limited (stripped/packed binary)
UNKNOWN	No analyzable content found

Q-Score Reference (YARA Rules)

Q-Score	Severity	Examples
5	CRITICAL	RSA, ECDSA, ECDH, DH, AES-128, MD5, SHA-1
4	HIGH	Weak PQC params, PRNG in crypto context
3	MEDIUM	Legacy crypto in quantum-aware systems
2	LOW	Performance-mode PQC, debug code
1	INFO	Migration recommendations

Reports Page

Select analyses from the Scan History table (latest per target shown), add optional context, and click **Generate Report**. Reports are saved to the database and available for download (Markdown or HTML) at any time. Delete reports you no longer need.

AI Features (Settings → AI & LLM Configuration)

Feature	What it does
AI Report Generation	Generates executive summary reports on the Reports page based on selected analyses and user context
AI Fix Generation	Suggests fixes for code findings — choose Comment annotation (safe, non-destructive) or Code generation (replaces vulnerable code)

Key Settings Locations

What	Where
License	Settings → License
Binary limits	Settings → Binary Analysis
Code scan extensions	Settings → Code Analysis
AI API key & features	Settings → AI & LLM Configuration (the key is kept in the OS credential store; `RIVET_AI_API_KEY` for the command line and CI)
Network port range	Settings → Network Analysis
Key Management connections	Key Management page (not Settings)
Log level	Settings → Application

Environment Variables

Variable	Purpose
`RIVET_LICENSE_KEY`	License key for headless/CI runs. Takes precedence over a key activated with `rivet enroll` or in the app.
`RIVET_SECRET_STORE`	`file` keeps the license key in a protected file instead of the OS keyring (containers, servers where the keyring prompts or is locked).
`OFFLINE_LICENSE_FILE`	Path to offline license file (air-gapped). Also a setting: `rivet config set license.offline_file <path>` (add `--scope machine` for every user of the machine). A file uploaded in the app is saved there automatically.
`LICENSE_SERVER_URL`	Address of your own license server. Also a setting: `rivet config set license.server_url <address>`.
`OFFLINE_PUBLIC_KEY`	Public key trusted to sign offline license files (needed with your own license server). Also a setting: `rivet config set license.offline_public_key <key>`.

Configuration (Command Line and Desktop App)

Priority, lowest to highest: defaults, machine file, user file, project file (`.rivet.yml`), environment (`LOG_LEVEL`, `LICENSE_SERVER_URL`, `OFFLINE_LICENSE_FILE`, `OFFLINE_PUBLIC_KEY`), flags. Settings an administrator lists under `locked:` in the machine file cannot be changed by anything else. The desktop app uses the same files: what you set in Settings → Code Analysis is saved to your user file, and a `.rivet.yml` in the scanned folder applies too.

Command	Purpose
`rivet config show`	Every setting in effect and where it came from
`rivet config path`	Where the machine, user and project files are
`rivet config validate [FILE]`	Check a file for mistakes
`rivet config set KEY VALUE [--scope user|project|machine]`	Change a setting
`rivet scan PATH --explain`	Which files would be scanned, which are left out and why (does not scan)

Files: machine `%ProgramData%\CerebionRivet\config.yml` (macOS `/Library/Application Support/CerebionRivet/`, Linux `/etc/cerebion-rivet/`); user `%APPDATA%\CerebionRivet\config.yml` (macOS `~/Library/Application Support/CerebionRivet/`, Linux `~/.config/cerebion-rivet/`); project `.rivet.yml`.

License Commands

Command	Purpose
`rivet enroll`	Activate this machine. Asks for the license key without showing it; also reads it from standard input or `--key-file FILE`, never from an argument. The key is kept in the OS credential store.
`rivet license status`	Show the license on this machine, without contacting the license server
`rivet license refresh`	Check with the license server now
`rivet license deactivate`	Release this machine's seat and remove the license from it

---

Troubleshooting Commands

Command	Purpose
`rivet doctor`	Check the installation. Each problem shows an error code and a next step. Add `--offline` on air-gapped machines.
`rivet diagnose -o support.zip`	Create a redacted support bundle to send to support@cerebion.com

---

Exit Codes (every scan command)

Code	Meaning
0	Clean: nothing was found, and everything was read
1	Findings: something quantum-vulnerable was found
2	Error: the scan could not run (bad options or credentials, license)
3	Incomplete: nothing was found, but part of the target could not be read, so this is **not** a clean result

`--no-fail` turns 1 into 0. `--allow-incomplete` turns 3 into 0. In a CI step, `rivet scan ./src --format json -o report.json` fails the build on findings by itself. Set `RIVET_LICENSE_KEY` in the pipeline's environment.

Log Locations

`rivet doctor` prints the exact log directory on your machine.

Platform	Path
Windows	`%APPDATA%\CerebionRivet\logs\app.log`
macOS	`~/Library/Application Support/CerebionRivet/logs/app.log`
Linux	`~/.local/share/CerebionRivet/logs/app.log`

Important Limitation

The Binary Analyzer performs **static analysis only**. A clean result means no known vulnerable patterns were detected — it does not guarantee the binary is quantum-safe. Runtime-only crypto, custom implementations, and encrypted/packed sections are outside detection scope.

Version 1.1.106 · **Support:** support@cerebion.com · **Docs:** cerebion.com/docs · **Portal:** cerebion.com/portal