

# chef

github.com/chef/chef

Scanned 24 September 2026 at 07:53 · Report generated 29 October 2026 · Ruby · 8,243 stars

OVERALL ASSESSMENT

Critical

Combined score 764 · 199 findings after filtering

## Summary

This repository has two distinct security issues that require immediate attention before it can be used in production. The most critical problem is a vulnerability score of 734, which indicates severe insecure coding patterns that an attacker could exploit to compromise the system. Additionally, there are medium-risk findings suggesting the code may contain malicious intent or suspicious behavior, such as downloading and executing remote files or using secrets improperly. While some issues appear to be ordinary mistakes like missing Rails security attributes or hardcoded keys, the presence of a YARA rule specifically designed to detect dangerous downloads combined with execution makes this repository unsafe for any development or deployment use.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

## Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

Critical (734)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

Medium (30)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

## What each detector found

| DETECTOR | LOOKS FOR                                      | HIGH | MEDIUM | LOW | TOTAL |
|----------|------------------------------------------------|------|--------|-----|-------|
| semgrep  | Insecure coding patterns across many languages | 56   | 106    | 0   | 162   |

| DETECTOR        | LOOKS FOR                                         | HIGH | MEDIUM | LOW | TOTAL     |
|-----------------|---------------------------------------------------|------|--------|-----|-----------|
| <b>yara</b>     | Signatures of malicious behaviour in source files | 18   | 0      | 9   | <b>27</b> |
| <b>guarddog</b> | Malicious package and install-time behaviour      | 0    | 10     | 0   | <b>10</b> |

162 of 199 findings were rated more likely genuine than noise by the classifier.

## Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. “Seen” is how many times that finding occurred.

| SEVERITY    | DETECTOR | FINDING                                                                                                                                                                                                                                                                                                      | SEEN | EXAMPLE LOCATION                                         |
|-------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------------------------------------------------------|
| <b>HIGH</b> | semgrep  | Found a potentially user-controllable argument in the construction of a regular expressions. This may result in excessive resource consumption when applied to certain inputs, or when the user is allowed to control the match target. Avoid allowing users to specify regular expressions processed by the | 2    | /spec/unit/encrypted_data_bag_item_spec.rb:425           |
| <b>HIGH</b> | semgrep  | Checks for models that do not use attr_accessible. This means there is no limiting of which variables can be manipulated through mass assignment. For newer Rails applications, parameters should be allowlisted using strong parameters. For older Rails versions, they should be allowlisted using strong_ | 42   | /spec/unit/chef_fs/data_handler/group_handler_spec.rb:22 |
| <b>HIGH</b> | semgrep  | A `run:` step pipes the output of `curl` or `wget` directly into a shell interpreter. This is the "curl   bash" install pattern — if the remote server is compromised or the URL is hijacked, an attacker can execute arbitrary code in your CI runner. Consider downloading the file first, verifying its c | 2    | /.github/workflows/func_spec.yml:157                     |
| <b>HIGH</b> | semgrep  | This workflow uses `secrets: inherit` to pass all of the calling workflow's secrets to a reusable workflow. This violates the principle of least privilege because the called workflow receives access to every secret in the repository, not just the ones it needs. If the called workflow is compromised  | 1    | /.github/workflows/ci-main-pull-request-stub.yml:33      |

| SEVERITY | DETECTOR | FINDING                                                                                                                                                                                                                                                                   | SEEN | EXAMPLE LOCATION                                                     |
|----------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------------------------------------------------------------------|
| HIGH     | semgrep  | JWT token detected                                                                                                                                                                                                                                                        | 1    | /spec/unit/secret_fetcher/azure_key_vault_spec.rb:35                 |
| HIGH     | semgrep  | Private Key detected. This is a sensitive credential and should not be hardcoded here. Instead, store this in a separate, private file.                                                                                                                                   | 7    | /spec/data/ssl/private_key_with_whitespace.pem:3                     |
| HIGH     | semgrep  | By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'. | 1    | /spec/integration/target_mode/support/docker/linux-ssh/Dockerfile:22 |
| HIGH     | yara     | YARA rule matched:<br>detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone                                                                                                          | 5    | /CHANGELOG.md                                                        |
| HIGH     | yara     | YARA rule matched:<br>detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this                                                                                                                     | 13   | /CHANGELOG.md                                                        |
| MEDIUM   | semgrep  | Detected non-static command inside 'open'. Audit the input to 'open'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.                                | 1    | /lib/chef/application/apply.rb:193                                   |
| MEDIUM   | semgrep  | Detected non-static command inside PTY.spawn. Audit the input to 'PTY.spawn'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.                        | 1    | /spec/functional/shell_spec.rb:90                                    |

| SEVERITY      | DETECTOR | FINDING                                                                                                                                                                                                                                                                                                      | SEEN | EXAMPLE LOCATION                            |
|---------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------|
| <b>MEDIUM</b> | semgrep  | Detected non-static command inside system. Audit the input to 'system'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.                                                                 | 1    | /spec/functional/resource/link_spec.rb:80   |
| <b>MEDIUM</b> | semgrep  | Should not use md5 to generate hashes. md5 is proven to be vulnerable through the use of brute-force attacks. Could also result in collisions, leading to potential collision attacks. Use SHA256 or other hashing functions instead.                                                                        | 19   | /spec/functional/http/simple_spec.rb:79     |
| <b>MEDIUM</b> | semgrep  | Detected SSL that will accept an unverified connection. This makes the connections susceptible to man-in-the-middle attacks. Use 'OpenSSL::SSL::VERIFY_PEER' instead.                                                                                                                                        | 8    | /lib/chef/monkey_patches/net-http.rb:104    |
| <b>MEDIUM</b> | semgrep  | The RSA key size 1024 is insufficient by NIST standards. It is recommended to use a key length of 2048 or higher.                                                                                                                                                                                            | 3    | /spec/unit/mixin/openssl_helper_spec.rb:265 |
| <b>MEDIUM</b> | semgrep  | Found the use of an hardcoded passphrase for RSA. The passphrase can be easily discovered, and therefore should not be stored in source-code. It is recommended to remove the passphrase from source-code, and use system environment variables or a restricted configuration file.                          | 4    | /spec/unit/mixin/openssl_helper_spec.rb:153 |
| <b>MEDIUM</b> | semgrep  | This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a `cooldown` block with `default-days: 7` to each `package-ecosystem` entry under `updates` to wait 7 days before proposing updates to newly published package versions. Reference: | 5    | /.github/dependabot.yml:59                  |
| <b>MEDIUM</b> | semgrep  | Detected a possible ZeroDivisionError.                                                                                                                                                                                                                                                                       | 3    | /spec/unit/mixin/user_context_spec.rb:82    |

| SEVERITY      | DETECTOR | FINDING                                                                                                                                                                                                                                                                                                      | SEEN | EXAMPLE LOCATION                                      |
|---------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------|
| <b>MEDIUM</b> | guarddog | Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]                                                                                                                                                                                                           | 3    | /chef-config/spec/unit/credentials_spec.rb:160        |
| <b>MEDIUM</b> | guarddog | Detects URLs to URL shorteners, file sharing, and suspicious services [threat-network-outbound-shady-links]                                                                                                                                                                                                  | 2    | /lib/chef/resource/windows_firewall_rule.rb:306       |
| <b>MEDIUM</b> | guarddog | Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]                                                                                                                                                                                                                      | 5    | /lib/chef/provider/user/solaris.rb:30                 |
| <b>MEDIUM</b> | semgrep  | Data is being piped into `bash` from a `curl` command. An attacker with control of the server in the `curl` command could inject malicious code into the pipe, resulting in a system compromise. Avoid piping untrusted data into `bash` or any other shell if you can. If you must do this, consider checki | 2    | /.expeditor/scripts/install-hab.sh:24                 |
| <b>MEDIUM</b> | semgrep  | GitHub Actions step uses a mutable tag or branch reference. Tags and branch names can be silently repointed by the action owner, enabling supply-chain attacks — as seen in the trivy-action and kics-github-action compromises. Pin the reference to a full 40-character commit SHA instead, e.g. `uses: ac | 58   | /.github/workflows/lint.yml:25                        |
| <b>MEDIUM</b> | semgrep  | The special variable IFS affects how splitting takes place when expanding unquoted variables. Don't set it globally. Prefer a dedicated utility such as 'cut' or 'awk' if you need to split input data. If you must use 'read', set IFS locally using e.g. 'IFS="," read -a my_array'.                       | 1    | /.expeditor/promote-docker-images.sh:126              |
| <b>LOW</b>    | yara     | YARA rule matched: detect_downloader — Downloads a remote resource - common and low-signal alone; see detect_download_and_execute for the composite version                                                                                                                                                  | 9    | /spec/data/cookbooks/wget/files/default/wget.response |

## How to read this report

---

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.
- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 24 September 2026.