CHECKLIST

Vulnerability Prioritization by CVSS Score Checklist

A practical, vendor-neutral checklist for analyzing CVSS results, validating exposure, and prioritizing vulnerabilities for remediation in a repeatable workflow.

Vulnerability Prioritization by CVSS Score

Checklist

Use this checklist to turn raw vulnerability findings into a defensible remediation queue. CVSS is a starting point, not the final decision.

---

1) Confirm you have the minimum inputs

- [ ] Vulnerability findings include CVSS base scores
- [ ] CVSS vector strings are available where possible
- [ ] Asset inventory includes owner, environment, and exposure details
- [ ] You can verify whether a finding is reachable or exploitable
- [ ] Scan data is current, trusted, and complete enough to act on

Stop here if any of these are missing. Do not prioritize from CVSS alone when you cannot identify exposed systems, production assets, or owners.

---

2) Normalize the findings

- [ ] Export findings into one table or ticket set
- [ ] Include these fields for every record:
- [ ] Asset identifier
- [ ] Vulnerability identifier
- [ ] CVSS base score
- [ ] CVSS vector string, if available
- [ ] Detected date
- [ ] Affected component or package
- [ ] Environment tag: prod, staging, or dev
- [ ] Asset owner
- [ ] De-duplicate repeated records by asset and vulnerability identifier
- [ ] Keep the most recent confirmation date
- [ ] Keep the highest-confidence record when duplicates conflict
- [ ] Verify every high or critical finding has an owner and environment tag
- [ ] Confirm scores are numeric and within the expected CVSS range

Output: one clean list of findings ready for ranking.

---

3) Bucket by severity band

- [ ] Sort findings by CVSS first-pass severity
- [ ] Group into bands:
- [ ] 9.0?10.0 = Critical
- [ ] 7.0?8.9 = High
- [ ] 4.0?6.9 = Medium

- [ ] 0.1?3.9 = Low
- [ ] 0.0 = Informational or no exploitable impact
- [ ] Within each band, sort by asset criticality and exposure
- [ ] Put public-facing production services ahead of isolated lab systems
- [ ] Sample items from each bucket and confirm the score matches the band

Output: a reviewable queue with clear severity buckets.

---

4) Check exploitability signals

- [ ] Review the CVSS vector string, if available
- [ ] Look for network reachability
- [ ] Check for low attack complexity
- [ ] Check whether privileges are required
- [ ] Check whether user interaction is required
- [ ] Look for known public exploit availability
- [ ] Look for active exploitation in your environment or sector
- [ ] Explain the likely attack path in plain language for a sample of findings

Priority increases when an issue is remotely reachable, easy to exploit, and already being weaponized.

---

5) Overlay asset context

- [ ] Mark whether the asset is production, staging, or non-production
- [ ] Mark whether the asset is internet-facing or internal-only
- [ ] Add system criticality or service tier
- [ ] Add data sensitivity or regulated-data impact

- [ ] Flag identity, authentication, management-plane, and control-plane systems
- [ ] Confirm exposure descriptions with the asset or service owner
- [ ] Correct inventory if ?internal-only? does not reflect real reachability

Remember: internal does not always mean low risk.

---

6) Check compensating controls

- [ ] Identify segmentation, filtering, or allowlisting controls
- [ ] Check for WAF, reverse proxy, or gateway protection
- [ ] Check for privilege separation or platform mitigations
- [ ] Verify controls are actually in place and correctly configured
- [ ] Confirm controls block the relevant attack path, not just the general system
- [ ] Ask for evidence instead of accepting assertions

Treat controls as risk reducers, not automatic closure reasons.

---

7) Apply a practical priority rule

Use a simple rule that blends CVSS with context:

- [ ] Start with CVSS severity
- [ ] Raise priority for internet-facing, production, or business-critical assets
- [ ] Raise priority for easy exploitation or active exploitation
- [ ] Lower priority for isolated, non-production, or strongly controlled assets
- [ ] Flag exceptions where a low CVSS item affects a high-value asset
- [ ] Flag exceptions where a high CVSS item is not urgent due to limited exposure

Goal: produce a queue that reflects actual remediation urgency, not just raw score.

---

8) Validate before assigning work

- [ ] Review a sample of top-priority findings with an analyst
- [ ] Review a sample with the asset owner or service owner
- [ ] Confirm exposure, exploitability, and asset criticality are correct
- [ ] Confirm the queue matches operational reality
- [ ] Fix inventory, tagging, or scanning gaps before publishing the final queue

Do not skip validation. A mechanically correct list can still be operationally wrong.

---

9) Assign remediation using a clear decision

For each finding, choose one outcome:

- [ ] Remediate now
- [ ] Remediate in scheduled maintenance
- [ ] Monitor and retest
- [ ] Accept risk with documented approval
- [ ] Reclassify after additional validation

Include a short reason for the decision, such as:

- [ ] Internet-facing production service
- [ ] Active exploit available
- [ ] Requires local access or user interaction
- [ ] Contained by verified compensating control
- [ ] Low-value non-production asset

Quick decision guide

Prioritize immediately when

- [ ] CVSS is high or critical and the asset is internet-facing
- [ ] The asset is production and business critical
- [ ] Exploitation is known, easy, or active
- [ ] The vulnerability affects identity, management, or control-plane systems

Defer or lower priority when

- [ ] The asset is isolated or non-production
- [ ] Exploitation requires unusual access or multiple preconditions
- [ ] Effective compensating controls are verified
- [ ] The finding has low operational impact in this environment

---

Common mistakes to avoid

- [ ] Treating CVSS as the only ranking signal
- [ ] Ranking by score without asset context
- [ ] Ignoring internet exposure
- [ ] Accepting stale or incomplete scan data
- [ ] Trusting undocumented compensating controls
- [ ] Assuming all 9.8 findings are equally urgent
- [ ] Overlooking low-score issues on high-value assets

Minimum output standard

Before you send the queue to remediation, make sure each prioritized finding has:

- [ ] Asset ID
- [ ] Vulnerability ID
- [ ] CVSS score
- [ ] Severity band
- [ ] Exploitability notes
- [ ] Asset context
- [ ] Control notes
- [ ] Owner
- [ ] Recommended action
- [ ] Review date

One-line summary

Use CVSS to start the ranking, then validate exploitability, exposure, asset criticality, and controls before deciding what to fix first.