THE WORKING TEMPLATE

Remediation Priority Template

A simple, defensible way to decide which vulnerability gets fixed first when the scan report is longer than the week. Score each finding on four factors, act in score order, and write the reasoning down.

How to use this template: a raw severity score answers "how bad is this weakness in the abstract". Remediation order needs a different question: "how bad is this weakness, on this asset, in this business, this week". The four factors below add that context. Score every finding, sort descending, and fix from the top. Where two findings tie, the one on the more exposed asset goes first.

The four factors

1 Exposure (0 to 3)

3: reachable by the whole internet. 2: reachable by many internal users or third parties. 1: reachable only by a small privileged group. 0: isolated or unreachable in practice.

2 Exploitability (0 to 3)

3: exploitation observed in the wild or trivially scriptable. 2: public exploit code exists. 1: exploitation is plausible but demands skill, access or luck. 0: theoretical only.

3 Asset criticality (0 to 3)

3: the asset gates the network, holds regulated or customer data, or moves money. 2: important to a business function. 1: inconvenient to lose. 0: disposable, test or decommissioned-in-name.

4 Compensating controls (subtract 0 to 2)

Subtract 2 where an existing control genuinely blocks the attack path (not merely logs it). Subtract 1 where a control raises the cost of attack meaningfully. Subtract nothing for controls you hope are working; if you have not tested it, it does not count.

Priority bands

SCORE	PRIORITY	WHAT IT MEANS IN PRACTICE
8 to 9	P1	Drop routine work. Fix, isolate or restrict now, and treat the asset as potentially compromised until patched.
6 to 7	P2	Into the current patch window; do not let it survive to the next scan cycle.
4 to 5	P3	Scheduled remediation alongside related work on the same system.
2 to 3	P4	Batch with routine maintenance, or formally accept the risk with the rationale recorded below.
0 to 1	Record only	Note it, watch it, spend no effort on it. Revisit if exposure or exploitability changes.

Worked example (fictitious findings)

FINDING	EXP.	EXPL.	CRIT.	CTRL.	SCORE	PRIORITY
VPN firmware with exploited vulnerability	3	3	3	0	9	P1
Exposed CMS admin login	3	2	2	-1	6	P2
Deprecated TLS on mail gateway	3	1	2	0	6	P2
Unpatched file server, internal only	2	2	2	-1	5	P3
Version disclosure in web headers	3	0	1	0	4	P3
Directory listing on marketing microsite	3	1	0	0	4	P3

Note the third row: two findings can share a score and a priority while deserving different dates. The tie-breakers, in order: exposure first, then exploitability, then whichever fix also closes other findings on the same system.

Your register

FINDING	EXP.	EXPL.	CRIT.	CTRL.	SCORE	OWNER	TARGET DATE
[Finding]							
[Finding]							
[Finding]							
[Finding]							
[Finding]							
[Finding]							

Accepted risks

FINDING	WHY THE RISK IS ACCEPTED	APPROVED BY	REVIEW DATE
[Finding]	[Rationale: no practical attack path, cost exceeds harm, planned decommission]		
[Finding]			

Keeping the register honest

RULES THAT KEEP THIS WORKING
Rescore when exposure changes: a system moved to the internet, a control removed, exploit code published
An accepted risk without a review date is a decision nobody made; every acceptance gets one
Findings older than two scan cycles at P2 or above go to whoever owns the budget, not just the backlog
Record what was fixed as well as what is open; the trend is your evidence for auditors and insurers

This scoring model is the one our consultants apply when triaging managed scanning findings, published so you can run it yourself. If you would rather the scoring, the register and the chasing simply happened every month, that is the service: managedvulnerabilityscanning.co.uk/pricing/.