Scenario: Who waits for the fix?

Andrew IDs:

The product and its customers

You work on the release team for **TinyNotes**. Your team operates its hosted web app, deploying tested 3.x builds before publishing them for customers to install.

Businesses also run a **self-hosted edition** on their own servers. Their administrators install updates and change settings; your team cannot do either remotely.

External sharing lets customers exchange documents with clients and partners through a browser or automate exchanges through the **sharing API**. Both routes should expose only documents the recipient has permission to access.

Your support commitments

Self-hosted customers run three major versions:

Version	Promised support	Share of installations
3.x	Full support	40%
2.x	Security fixes for six more months; keep existing integrations working	50%
1.x	Support ended three months ago; no routine fixes	10%

The security issue

A security researcher has discovered a major security issue and will publish it in 48 hours.

- Anyone who can reach the sharing API can download private documents **without signing in**, even documents never shared externally. The flaw does not allow changes or deletions.
- All three versions and the hosted service are affected; the hosted fix is not deployed. Public servers are reachable by anyone; private servers require network access.
- Your team has seen **no evidence of attacks**, but logs cannot reliably distinguish them from legitimate downloads. You cannot inspect customer logs remotely.

Key terms

- **API:** An interface that lets software send requests to another system.
- **Integration:** A connection that lets two systems work together, such as automatically creating employee accounts.
- **Regression testing:** Checking whether a change breaks behavior that previously worked.
- **Mitigation:** A temporary measure that reduces exposure without fixing the underlying flaw.

Patch options

3.x — Ready to release

The fix has passed all required checks. You can deploy it to the hosted service and publish it for customers to install.

2.x — Two options; neither is ready to release

Option A: Finish the adapted 3.x fix. The prototype blocks the attack but breaks the integration that automatically creates TinyNotes accounts for a customer's new employees. Both features use authentication code changed by the fix. Existing accounts and document sharing still work.

Tests for other regressions are incomplete. Engineers have not estimated the remaining checks and release preparation, so **availability today is not guaranteed**.

Option B: Preserve existing integrations. A compatible fix needs **2–4 days**, possibly longer.

1.x — Not investigated

No patch or time estimate is available.

Upgrading to 3.x is not immediate: older versions require data migration and integration changes.

Temporary mitigations

Administrators can use these mitigations or others you propose. Explain what each protects, what stops working, and what needs checking.

- **Turn off external sharing.** Blocks the attack and disables both browser sharing and automated document exchange; internal notes still work.
- **Block the sharing API.** Blocks the attack and automated document exchange. Browser sharing uses separate endpoints unaffected by the flaw and remains available. A gateway sits in front of the server and filters incoming requests. Administrators must use the tested gateway setup and ensure requests cannot bypass it.
- **Disconnect the server.** Blocks the attack and all network access.

Version 1.x has no sharing switch, and the gateway rule has only been tested on 2.x.

> **Your team cannot do everything at once.**
>
> Work on 1.x, the 2.x prototype, or other gateways delays the compatible 2.x fix. Testing other gateway configurations on **2.x** adds **one day**. Validating the gateway rule on **1.x** has no time estimate. The ready 3.x fix can ship independently.

Your plan

Plan your response over the next **48 hours** for the hosted service and customers on **3.x, 2.x, and 1.x**.

Use brief bullets; you may refer to earlier answers.

1. What will you release or deploy today? Why?

2. What work comes next? Why this order?

3. What should administrators do today?

Your plan (continued)

4. What tests are still needed?

5. Who remains at risk?

6. What stops working?

Additional notes and revisions after the customer request