Rublon

Rublon Business Continuity Preparedness Guide

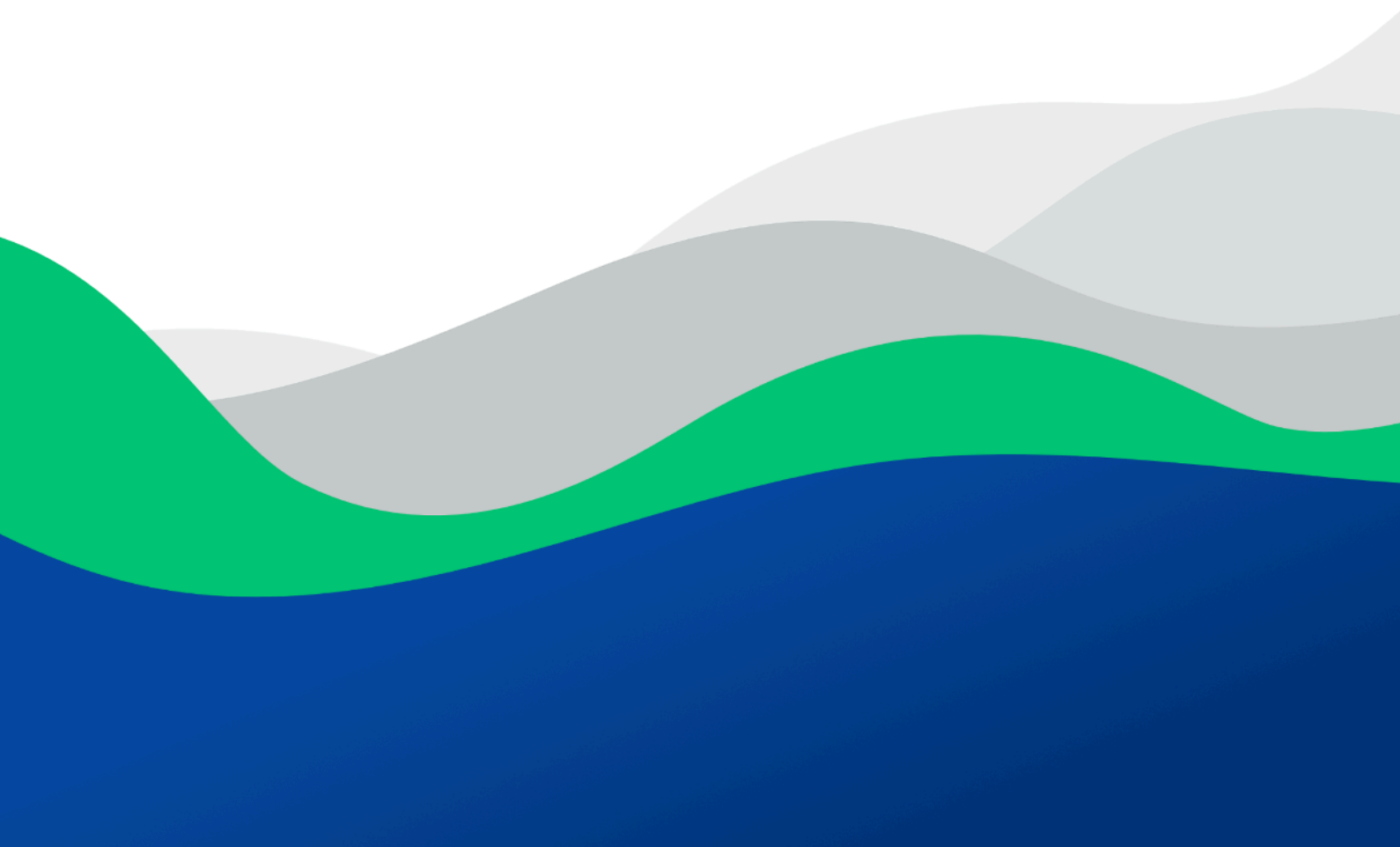

Table of Contents

1. Overview

Why You Need This Guide:

Even the most robustly engineered services can occasionally face disruptions. Rublon MFA is designed for high availability (with a track record of over 99.9% uptime), but no cloud service is immune to outages.

A brief period of unavailability – however rare – could impact your users' ability to access critical systems. Such outages may temporarily disrupt workforce productivity and even weaken your security posture if not handled properly.

As your trusted multi-factor authentication provider, Rublon wants you to be prepared for any situation. This guide will help you formulate a plan to maintain secure access during Rublon service outages or degradations. Plan ahead to ensure your organization continues operating smoothly even if Rublon's MFA service encounters an issue.

Scope:

This guide focuses on continuity strategies for Rublon MFA. We will explain:

- Different types of service disruptions
- How Rublon-integrated applications behave in each scenario
- What configuration choices and workarounds can you employ

The guide also provides example communications to keep your end-users informed during an incident. With this knowledge, you can create a tailored contingency plan for your organization's needs.

2. Is It Rublon or Your Firewall?

What appears to be a Rublon outage can be caused by environmental factors like network configuration.

Consider how your Rublon-protected applications connect to the Rublon API and what dependencies exist. For instance, if your organization has strict firewall rules or filters specific traffic, you must ensure Rublon's service is allowed through. Rublon runs on Amazon Web Services and uses dynamic IP addresses that can change over time. If your firewall is not updated to allow these addresses or the Rublon API endpoint, it might block Rublon requests, resulting in authentication failures even though Rublon's service is up. (In other words, an internal network misconfiguration can **mimic** a Rublon outage.)

To prevent future issues, make sure your environment is prepared: configure your firewall for Rublon. Also, ensure standard requirements like DNS resolution and TLS 1.2+ support are in place so that all systems can reach Rublon's servers.

Taking the preceding steps now will prevent many connectivity issues caused by environmental blockers and help ensure Rublon can consistently communicate with your applications.

3. Preparing for Outages

Preparation is key to minimizing the impact of any outage. Once you understand the possible outage scenarios and how each of your applications behaves (with or without Rublon), we highly recommend creating **application-specific disaster recovery (DR) plans**.

Each Rublon-protected system may require a slightly different continuity procedure. Planning these details ahead of time ensures you can respond swiftly and confidently if an outage occurs. The goal is that **no one is caught off guard** – administrators will know what to do, and users will know what to expect.

As you develop your plans, consider the following:

- **Failover/Bypass Procedures:** Document the process to manually bypass Rublon MFA or "fail open" an application if the Rublon service is down and automatic failover is not working. For example, this might involve changing a setting, updating a registry key, or rerouting authentication to temporarily bypass Rublon MFA. Know how to quickly invoke these bypass options and how to set Fail Mode on each Rublon connector.
- **Encourage Multiple Enrolled MFA Methods:** As part of your standard onboarding process, ask each user to enroll at least two MFA methods (e.g., mobile push, TOTP, SMS, email, and FIDO security keys). This ensures that if one method experiences an outage, users can switch to another method without creating internal helpdesk tickets.
- **Removal/Disable Steps:** Outline how to remove Rublon from the authentication flow for each protected application if absolutely necessary. This could mean uninstalling or turning off a Rublon connector, disabling an MFA enforcement setting in the Rublon Admin Console, or reverting to a local login method. Identify the steps and necessary privileges in advance so that an administrator can perform them under pressure.
- **Responsible Personnel:** Determine who in your organization is responsible for ensuring business continuity. Ensure they have access and credentials needed to make configuration changes (e.g., access to servers where Rublon connectors are installed, an administrator account in the Rublon Admin Console, etc.). It's a good idea to have more than one person familiar with the plan in case the primary responder is unavailable.
- **Testing and Drills:** Don't wait for a real incident to find out if your plan works. Whenever possible, test your failover procedures in a controlled way (e.g., simulate a Rublon outage

in a staging environment). Regular drills will validate that your documented steps allow users to log in when Rublon is unreachable and help staff become comfortable performing them. If something in the plan does not work during a test, update the documentation accordingly.

- **Communication Plan:** Decide how you will communicate with IT teams and end-users during an outage. (We provide example user communication templates later in this guide.) Having pre-written messages and an internal notification process will save time when every minute counts. Additionally, ensure that you prepare both primary and backup communication channels, such as email notifications or dedicated chat channels, so that if the main channel fails, a reliable alternative is available. Establish a robust process for delivering messages, along with backup procedures, to further enhance your outage preparedness.

4. Configuration Decisions: Fail Safe vs. Fail Secure

A critical part of continuity planning is deciding how each application should behave if Rublon becomes unavailable. Most Rublon connectors allow you to configure a **Fail Mode**, essentially choosing between a "**fail open**" (allow access) and "**fail closed**" (deny access) approach when MFA cannot be completed. We sometimes refer to these as **Fail Safe** (bypass) and **Fail Secure** (deny) modes. Deciding which mode to use for each application is a balance between security and usability, and it should align with your organization's policies. See Section 7 for a detailed explanation of Fail Mode behaviors.

Most Rublon connectors allow you to configure the Fail Mode behavior. In the next sections, we will delve into what happens in each type of outage and how Fail Mode comes into play. Keep in mind which mode you have set (or will set) for each application, and document those decisions, as they directly affect your outage handling plan.

Consider the following factors when choosing the Fail Mode for an application:

- **Policy and Compliance Requirements:** Do any compliance regulations or internal security policies mandate that access must be blocked if MFA cannot be verified? For example, some standards require MFA on specific systems at all times. If so, Fail Secure (deny access without MFA) is necessary for those systems. On the other hand, less regulated environments might permit using Fail Safe (bypass MFA) to maintain operations. The decision should be documented and periodically reviewed as regulations and organizational priorities change.
- **Sensitivity of Data and Systems:** Evaluate the type of data and resources the application protects. For systems containing highly sensitive information (financial records, patient data, critical infrastructure controls, etc.), you should favor security over availability, leaning toward Fail Secure to decrease the risk of unauthorized access without MFA. For applications with more general, low-risk data, you can prioritize availability via Fail Safe so users can continue working during an outage.
- **User Groups and Access Levels:** One size may not fit all. You might have a mix of users with different risk profiles. For instance, users with privileged access could be required to use Fail Secure (no bypass), whereas general staff accessing less critical systems might be allowed to use Fail Safe.

- **Balancing Security and Usability:** Consider the impact on your users if they are denied access vs. the impact on security if MFA is bypassed. Fail Safe (bypass) ensures users never lose access to their systems due to an MFA outage. This improves uptime and productivity, but at the cost of temporarily not having the second authentication factor. On the other hand, Fail Secure maintains the MFA requirement at all times, preserving security, but potentially at the cost of users being unable to log in until the MFA service is restored. You may choose Fail Safe for systems where continuous access is paramount (e.g., Windows endpoints), and Fail Secure for systems where an MFA bypass is an unacceptable risk (e.g., VPN access to a production network).

5. Types of Outages

Not all service interruptions are the same. Rublon MFA service disruptions fall into two broad categories:

- **Rublon API Unreachable** - A complete outage where Rublon connectors cannot contact the Rublon API. Typically triggers the connector's configured Fail Mode (Fail Safe or Fail Secure), automatically allowing or denying logins.
- **Rublon Service Degradation** - A partial service disruption where the Rublon API is still reachable, but specific Rublon services are slow, failing intermittently, or unavailable. Fail Mode is not triggered automatically, so admins may need to intervene manually.

Why Distinguish Unreachable vs. Degraded?

The main reason is that your response will differ. If Rublon is completely unreachable, much of your continuity plan will rely on the **Fail Safe or Fail Secure settings** you put in place beforehand. In contrast, if the service is degraded, you might need to **actively intervene** (for example, by communicating a workaround to users or temporarily changing a setting) because the system will not automatically fail over. Keep these categories in mind as we move forward, and consider how each of your applications should behave in each scenario.

Rublon API Unreachable

This scenario is a complete interruption of connectivity to the Rublon API. Essentially, your Rublon connectors **cannot contact the Rublon API at all (HTTP Code 404), or contact has been made, but an HTTP Code 500-599 was returned**. This scenario invokes the Fail Mode.

In a scenario where the Rublon API is unreachable, the Rublon Prompt (the screen that prompts for the second factor) will not load during login, or the request to the Rublon API will time out for promptless integrations. Rublon connectors typically wait a certain amount of time for a response before deciding the Rublon API is unreachable. After that determination is made, the connector's **Fail Mode** takes over, automatically allowing or denying user logins based on your chosen setting. We often call this a "failover" scenario since the connector is failing over to its predefined mode (safe/bypass or secure/deny).

Causes of an “unreachable Rublon API” event might include:

- **Rublon Downtime:** The Rublon API is temporarily down or not responding. This is rare but can happen (e.g., due to planned maintenance).
- **Network Connectivity Issues:** Problems in the network path between your environment and the Rublon API can render the service unreachable. These issues often originate on the customer side or within the internet infrastructure. Examples include an ISP outage or routing problem, a DNS failure resolving Rublon’s domains, and an accident (like a fiber cut) taking down connectivity. Local misconfigurations are also common – a firewall rule change or proxy issue could suddenly start blocking traffic to Rublon, making it appear unreachable.
- **Misconfiguration of Integrations:** If a Rublon connector is set up incorrectly, it might not be able to reach Rublon’s API. For example, an incorrect Rublon System Token or Rublon Secret Key in the configuration can cause authentication calls to fail. In this case, the connector might treat it similarly to an unreachable service because it never gets a valid response. Similarly, using an outdated Rublon connector version that is incompatible could result in errors contacting the service.

Rublon Service Degradation

This refers to a partial service issue where the Rublon API is **still reachable**, but some aspects of the MFA service are not functioning properly. In a degradation, your applications can contact Rublon (so connectors do not trigger the Fail Mode), but users may experience errors or delays in completing MFA.

In a degradation scenario, **connectors do not automatically invoke the Fail Mode** because the Rublon API is still responding (even if with delays). People responsible for business continuity must recognize the signs of a degraded service (e.g., multiple users reporting that their SMS codes or emails are not working) and then decide if they need to take action, such as manually bypassing MFA for the duration of the issue.

Examples of degradation include:

- **Specific Method Failures:** One or more of the second-factor methods is not working, while others are fine. For instance, perhaps Rublon's telephony provider for SMS or phone calls is experiencing an outage, so SMS codes are not being delivered to users. Meanwhile, other methods like Mobile Push or FIDO security keys still work.
- **Partial Outage or Slowness:** Rublon's infrastructure might be under heavy load or experiencing a localized issue that increases authentication latency. Users might find that their second-factor prompts are slower than usual or intermittently failing, but eventually, requests go through. This could occur if one component of Rublon's cloud (e.g., a particular server cluster) is having problems while others continue to operate. Users may report that "MFA is hanging" or that they have to attempt to log in multiple times to succeed.
- **Software Issues:** In extremely rare cases, a software bug can degrade the service. For example, if a new release introduces an error, some authentication attempts might not complete successfully (even though the service is up). While we hope such events never occur, it is wise to be aware that they are possible – the service is technically "there" but not fully functional.

6. Detecting Outages

How do you know when an outage is occurring? Early detection is critical. By staying vigilant with these monitoring steps, you will quickly detect an outage and distinguish its nature (unreachable vs. degraded). Quick detection means you can activate your continuity plan faster.

Here are the steps and tools to help you quickly recognize a Rublon service issue:

- **Monitor Rublon's Status Page:** Rublon provides a real-time Status Page at status.rublon.com. This page should be your first stop when you suspect any widespread issue. The Status Page reports the health of core components like the Rublon API, Admin Console, Support Portal, Website, and even specific features like SMS/Phone/Email delivery. If we become aware of a service disruption or schedule upcoming maintenance, we will post updates on our Status Page. For that reason, we recommend you subscribe to updates. This way, you will be proactively notified if an incident arises. Subscribing ensures you never miss critical announcements about outages or maintenance.

- **Local Diagnostics:** If you are experiencing issues but the Status Page shows all Rublon services operational, the problem might be on your side or a network in between. Here are a few diagnostic steps:

    - **Check Connectivity to Rublon API:** Try to connect to Rublon's API endpoint from your environment. For example, open a web browser on a machine that's having trouble and navigate to **https://core.rublon.net**. This URL is Rublon's core authentication server. If it is reachable, you should see a simple message like "Rublon Authentication Server works!" in the browser. Perform this test from multiple networks and machines. If one network shows the "works" message and another does not, that suggests a network issue (like a firewall or routing problem) on the failing side. This test can help pinpoint whether Rublon is truly down or if something in your environment is blocking access.

    - **Traceroute/DNS:** Standard network tools can help identify issues. Use a tool like **tracert/traceroute** to see if traffic is reaching the internet. Also, verify that the DNS resolution (**nslookup core.rublon.net**) returns an IP address. If DNS fails or the trace dies inside your network, you have likely found a local issue.

    - **Firewall/Proxy Logs:** If you have a web proxy or firewall, check its logs around the time of the failures. You may discover that connections to Rublon's domains/IPs

are being dropped or denied. Rublon's IP addresses can change (due to AWS hosting), so an updated rule might be needed. Look especially for any recent firewall changes and security software (like IDS/IPS) alerts that coincide with the outage.

- **Differentiate Widespread vs. Isolated Issues:** Determine if the problem affects all users and applications or just certain ones. If **all authentications are failing** across the board, that points to a broader outage (unreachable Rublon API or major network break). If only a specific method or application fails, it might be a degraded service or app-specific issue. For instance, if only SMS codes are not arriving but push and email are fine, that's likely a Rublon degradation in the SMS channel (or an SMS provider issue). If only your VPN (using Rublon Authentication Proxy) is having issues, but Windows logins with Rublon are fine, the issue could be isolated to the proxy or its configuration.
- **Contact Rublon Support:** If you have gone through the above and are still unsure or need assistance, don't hesitate to reach out to Rublon Support. In your support request, include as much detail as possible (what's failing, error messages, screenshots, and any log excerpts).

7. Rublon Fail Mode

Rublon connectors include a concept called **Fail Mode** that controls what the connector should do if it fails to complete MFA.

The possible values of Fail Mode are:

- **Fail Safe (Bypass)** - if the Rublon API becomes unreachable, users will be **allowed access** to Rublon-integrated applications if they pass primary authentication.
- **Fail Secure (Deny)** - if the Rublon API becomes unreachable, users will be **denied access** to Rublon-integrated applications even if they pass primary authentication.

Both Fail Safe and Fail Secure have their place. Rublon gives you the flexibility to decide per connector which mode to use. It's even possible to mix modes: e.g., set most connectors to Fail Safe, but a few highly sensitive ones to Fail Secure. The key is to configure these ahead of time and document your choices so that you are always ready for planned maintenance or an unplanned outage.

Fail Safe Mode ("Bypass When Unreachable")

Fail Safe means that if the Rublon API is unreachable, **the user can log in without completing MFA**. In other words, the second-factor step is skipped (bypassed), and the primary credential (usually username/password) alone will grant access.

- **Priority:** Fail Safe prioritizes availability. Users can continue working even if the MFA service is down. However, Fail Safe comes at the cost of reduced security during the outage (since users are not challenged with a second factor).
- **Behavior:** From the end-user's perspective, a Fail Safe event looks like a usual login, except they are never prompted for their 2FA. For example, an employee might log in to their Windows laptop as usual, enter their username and password, and then, instead of seeing the Rublon Prompt, the login just completes. This indicates the system bypassed the user.
- **When to Use:** Fail Safe is generally recommended for systems where maintaining access is more important than enforcing MFA at that moment. Many organizations choose Fail Safe for user-facing systems like workstation logons or internal applications to avoid

locking out staff. If you implement Fail Safe, ensure you have other controls in place (like strong primary authentication, network monitoring, robust access control, etc.) to mitigate the temporary loss of MFA during an outage.

- **Configuration:** Each Rublon connector has its own way of setting Fail Safe (usually by setting the **FailMode** option to "**bypass**"). **Rublon connectors default to Fail Safe** out of the box to avoid admins locking themselves out during initial deployment. To change this behavior for each connector, refer to How to set Fail Mode in Rublon connectors?. If you decide to keep Fail Safe, it is wise to test the Fail Safe behavior (e.g., by temporarily blocking the connector's internet access in a test environment) to confirm it indeed bypasses as expected.

Fail Secure Mode ("Deny When Unreachable")

Fail Secure means that if the Rublon API is unreachable, the **user will be denied access**. In this mode, a user who has correctly entered their primary credentials will still be **blocked from access if the second factor cannot be verified**.

The impact of Fail Secure is straightforward: no Rublon API = no login. It maximizes security at the expense of continuous access. If you choose this for any application, make sure your **business continuity plan addresses how to handle the downtime**. That could be informing users to wait or using a workaround like an alternative login method (e.g., an emergency local account that isn't under Rublon, though that introduces its own risks). In many cases, organizations use Fail Secure on systems with limited user bases or where an outage, while disruptive, is acceptable compared to the risk of unauthorized entry.

- **Priority:** This mode prioritizes security – it never lets someone in without MFA – but at the risk of making the application unavailable during an outage (since even legitimate users will be locked out until MFA is working again).

- **Behavior:** In a Fail Secure scenario, users will be denied access if a connection to the Rublon API cannot be made. After entering the username/password, Rublon will immediately show a message like "Access Denied!". This can be frustrating if the user does not anticipate it. They have done nothing wrong but are denied access due to the MFA setting. This is why informing users about Fail Secure during an outage is of paramount importance.

- **When to Use:** Fail Secure is appropriate for high-security systems where denying all access is preferable to allowing logins verified only by primary authentication. If an application safeguards sensitive data or serves as a gateway to a larger network, you might decide it should always require MFA, even if it means accepting downtime during planned Rublon maintenance. Some organizations set Fail Secure on externally facing entry points like VPNs or privileged IT administration portals. Before choosing Fail Secure, make sure you weigh the impact:
    - How critical is it that this system remains accessible?
    - Are there alternative ways to get into the system if MFA is down (e.g., a break-glass local admin account, or console access)?
    - Will users know that an MFA outage means “please wait, you can’t get in right now”?
- **Configuration:** Enabling Fail Secure involves toggling the Fail Mode setting to “deny”. For detailed instructions for each connector, refer to How to set Fail Mode in Rublon connectors?. It is wise to test the Fail Secure behavior (e.g., by temporarily blocking the connector’s internet access in a test environment) to confirm it indeed denies access as expected.

8. Fail Mode Support by Rublon Integration

Rublon applications, plugins, and connectors that **provide Fail Mode control**:

- Rublon Authentication Proxy
- Rublon Access Gateway
- Rublon MFA for Windows Logon & RDP
- Rublon MFA for Remote Desktop Gateway
- Rublon MFA for RD Web Access
- Rublon MFA for RD Web Client
- Rublon MFA for Outlook Web App (OWA)
- Rublon MFA for Active Directory Federation Services (AD FS)
- Rublon MFA for Linux SSH
- Rublon MFA for Veritas NetBackup
- Rublon MFA for Jira
- Rublon MFA for Confluence

Rublon integrations that **do not provide Fail Mode control**:

- Rublon MFA for Roundcube
- Rublon MFA for WordPress

For detailed instructions for setting the Fail Mode on each connector, refer to:

How to set Fail Mode in Rublon connectors?

9. Rublon API Unreachable Scenarios and Responses

The Rublon API may be unreachable in different ways: it can be truly down for everyone (cannot be reached from any network or endpoint), or it can be inaccessible for only some connectors or locations (often due to a firewall or network misconfiguration). In either case, the connector sees no valid response and applies its Fail Mode.

Communication and monitoring are vital:

Keep your IT security team and management in the loop about what temporary fixes you are applying. For instance, if you have set all finance users to bypass MFA, let the security officer know. There might be a trade-off in risk that they need to sign off on. After the fact, log or report what was done (e.g., “On 10:00 AM, due to Rublon outage, set VPN to bypass mode”). This helps in post-incident reviews and audits

Let your help desk and users know you are aware of the issue and what kind of Fail Mode setting you use:

- If you use **Fail Safe (bypass)**, users might be surprised or even worried that they are not being challenged for MFA.
- If you use **Fail Secure (deny)**, users might be surprised that they cannot access their applications.
- A quick message can save a lot of confusion. We provide sample user messages later on in this guide.

Example: The Rublon API is unreachable. It cannot be reached from any network or endpoint. The Rublon API is unreachable for all Rublon connectors.

Impact:

- If Fail Secure is set, users cannot complete MFA.
- If Fail Safe is set, users are allowed access (without MFA).

Detection: The Rublon Status Page will show issues with the Rublon API. If the Rublon Status Page does not show any issues with the Rublon API, this strongly suggests a connector configuration issue or a misconfigured firewall.

Response Options:

- **Switch to Bypass (Fail Safe) Mode:** If you normally run an application in Fail Secure mode, one quick workaround during an outage is to flip it to Fail Safe. This could be done by changing the Fail Mode to “bypass”. For example, if the Rublon Auth Proxy is set to **“deny”**, edit the config file, set **fail_mode: "bypass"**, and restart the proxy service, effectively allowing users in without MFA until you revert it. Many administrators do this proactively if they see an outage coming (for instance, when we announce scheduled maintenance, you might pre-set everything to bypass to avoid any hiccups. **Keep track of where you made these changes so you can revert afterward. You do not want to accidentally leave an application in permanent bypass mode because that becomes a security gap.**
- **Adjust Firewall/Network:** If you discover that the issue is due to your firewall or network blocking access to the Rublon API (for example, a new firewall policy prevents access to **core.rublon.net**), then the workaround is to fix that network issue as soon as possible. This might involve:
    - Opening outbound port 443 and adding **https://core.rublon.net** to the allowlist.
    - Adding the latest Rublon AWS IP ranges to your firewall.
    - Temporarily disabling a restrictive rule (with proper approvals) until you can refine it to allow Rublon.
    - Learn more: How should I configure my firewall for Rublon?

10. Degraded Service Scenarios and Responses

When Rublon's infrastructure is partially degraded, you may need to take additional steps to minimize disruption. Unlike a full outage, a degraded scenario means MFA is still functioning to some extent, so automatic failover (Fail Mode) **will not trigger**.

Communication and monitoring are vital:

Let your help desk and users know you are aware of the issue and what workarounds to use. Users encountering MFA trouble might keep retrying or assume it's their device. A quick message can save a lot of confusion. We provide some sample user messages later on in this guide.

If degradation is specific to one connector or integration:

For example, only MFA for your VPN is failing, while other apps with Rublon are fine. In that case, your continuity plan might involve failing over that single application (e.g., temporarily disabling Rublon on the VPN until you fix the integration). Always determine if the issue is global or isolated, as the response will differ.

Scenario A: Specific Authentication Method Outage

Example: Rublon's SMS delivery service is down, and passcodes via SMS are not reaching users. Other methods (push notifications, mobile app TOTP codes, email links, phone calls) are working.

Impact: Users who rely solely on the affected method (SMS in this example) will be unable to complete MFA. They might request the SMS code and never receive it. Users using other methods won't be affected.

Detection: Rublon's Status Page will show an issue with SMS delivery (look at the components list – *"SMS Message Delivery"* will be marked degraded). Users will complain, "I am not getting my code."

Response Options:

- **Suggest an alternative authentication method:** Inform users about the issue and advise using an alternate authentication method, like Mobile Push or Email Link. This requires that users have another authenticator enrolled. Ideally, all users should be encouraged to

have at least two authenticators (for example, Rublon Authenticator + phone number, or FIDO2 security key + third-party app) so they have a backup.

- **Use a Bypass Code:** If no other authentication method is feasible and a given user has to access a resource, an admin can generate a Bypass Code (a one-time code that skips MFA) and instruct the user on how to use Bypass Codes.
- **Monitor the status:** Monitor the Rublon Status Page for updates. If you subscribed to the Status Page, you will receive an automatic email message when the issue is resolved. These types of single-method outages are usually resolved quickly by the third-party provider.
- **Do not change Fail Mode:** In this scenario, there's typically no need to change the Fail Mode, as MFA authentication remains operational. The focus should be on providing clear guidance to users.

Scenario B: General Service Slowness or Partial Failure

Example: Rublon API is responding slowly, or some auth requests are intermittently failing. Users report that sometimes their MFA prompt times out or they get errors, but after a few retries, it works.

Impact: Authentication is unreliable. This can be frustrating for users and could slow down their work. It may not completely stop access, but it creates delays and confusion.

Detection: Users will report *"MFA is not working consistently"* or "MFA is slow." The Rublon Status Page will show a degradation of one or more services.

Response Options:

- **Check the Rublon Status Page:** Degradation issues are resolved relatively quickly (often within minutes to an hour). Plan for how long you are willing to operate in a degraded state before you escalate your response (e.g., enact the bypass plan after 30 minutes of high failure rates).
- **Reassure users:** If the issue is minor and the status page indicates a quick fix, the best approach might be to notify users that "MFA is currently experiencing intermittent issues, please retry if it fails, and expect possible delays." Sometimes, just setting expectations helps reduce helpdesk tickets.

- **Temporarily Bypass MFA:** If the degraded state of a service is seriously impeding business (logins are taking too long and work is stopped), you can invoke a bypass until the issue is resolved:
    - Change all or select users' status to **Bypass**.
    - Bulk add users to a group with a **Bypass** status.
    - Enable **Fail Safe** (FailMode=bypass) in one or more Rublon connectors and create a firewall rule to block **core.rublon.net** on TCP port 443 to enforce MFA bypass.
    - If absolutely necessary, temporarily disable or uninstall the Rublon connectors (**not recommended**).
    - Ensure that any changes are communicated to the team so everyone knows MFA is intentionally bypassed, and **don't forget to roll back the changes** after the incident is resolved.

11. End-User Messaging Templates

Clear communication with end-users is vital during an outage or service disruption. Users may panic or flood the help desk with tickets if they suddenly can't log in or their MFA prompts behave differently. Having pre-written messages allows you to quickly inform your users about what's going on and what (if anything) they should do. Below are some friendly, professional templates you can adapt and send out. You can send these via email, chat, or whatever communication channel is appropriate for your organization.

Deciding when to notify users:

Before using the templates, consider when to send them. You might decide to notify users as soon as an issue is confirmed (especially if it affects a large portion of users and critical systems). Or you might wait a few minutes to see if it resolves quickly (to avoid unnecessary alarm), especially if it's after-hours or affecting only a small subset of users. Use your judgment based on the time of day and criticality (e.g., an issue at 9 AM on Monday needs swift communication; an issue at midnight on Saturday for a system used only on weekdays might not require an immediate alert). Also, if you expect the issue to be resolved very shortly (say, the Rublon Status Page indicates a one-minute maintenance), you might opt to hold off unless it extends.

Template 1: General Outage – Fail Safe (Bypass)

Scenario: The Rublon API is unreachable, or one or more Rublon services are degraded. You have enabled MFA bypass (Fail Safe) as a workaround so users can log in with just their primary authentication for now.

Subject: Authentifizierungsprobleme – MFA vorübergehend deaktiviert

Body:

> Our multi-factor authentication provider (Rublon) is reporting an issue with its services. To ensure everyone can continue working, we have temporarily **lifted the requirement for Rublon MFA** at login. This means you will not be prompted for the usual second factor right now, and you'll log in with just your username and password. Please continue to be vigilant about security during this time. After the issue is resolved, we will **reinstate the MFA requirement**. We'll update you when that happens.

Template 2: General Outage – Fail Secure (No Bypass)

Scenario: The Rublon API is unreachable, or one or more Rublon services are degraded. You are **NOT bypassing MFA** (Fail Secure mode remains in effect). Users will be denied access until the issue is fixed.

Subject: Authentication Issues - Access Temporarily Unavailable

Body:

> Our multi-factor authentication provider (Rublon) is reporting an issue with its services. As a result, logins to **<affected systems>** are not possible at this time. This is a deliberate precaution due to the sensitive nature of those systems - we cannot grant access to these systems without MFA. We understand this is frustrating, and our IT team is working closely with Rublon to resolve the issue as soon as possible. **Please do not attempt repeated logins until we give the all-clear, as they will not succeed anyway.** We will provide an update as soon as the issue is resolved or an alternative access method becomes available.

Template 3: Authentication Method-Specific Issue

Scenario:

A specific MFA method is having issues (but others work). For instance, email authentications are failing, but other methods (SMS, Phone Call) work. Another example: SMS delivery is significantly delayed.

Subject: MFA Service Advisory - Use Alternate Authentication Method

Body:

> Our multi-factor authentication provider (Rublon) is reporting an issue with **<affected method>**. This means if you normally use **<push/SMS/phone>** for the second authentication step, you may have trouble completing the login. As a workaround, please use **<alternate method>**. We're monitoring the situation and will let you know when the **<affected method>** is working normally again. If you're unsure how to use an alternate method or still cannot log in, contact the IT service desk for assistance.

Additional Communication Tips

- Send follow-up messages when things are resolved: e.g., *"The MFA service issue has been resolved. Rublon MFA is fully operational again. If you switched to a backup method, you can return to your preferred method now. Thank you for your patience."*
- Keep the tone calm and helpful. During outages, users might be anxious (*"I can't log in to work, what now?"*). Assure them IT is on it, and provide any actions they should take (or not take).
- If only a subset of users is affected (such as a specific office or group), tailor the message to that audience to avoid confusion.
- If the outage is extended or the workaround changes, update users periodically (e.g., *"MFA service is still down as of 11:30 AM, we are continuing to bypass it for now. Next update in 1 hour or when we have news."*).

12. Post-Incident Review

After any Rublon outage or service degradation, it's good practice to hold a short review session with your IT or security team. This helps identify what worked well and what could be improved next time.

Questions to consider:

- Were users able to access systems as expected under the configured Fail Mode?
- Did admins apply the appropriate response steps, such as switching to bypass or using alternate methods?
- Was communication to users clear and timely?
- Were any security exceptions (e.g., temporary bypass) properly logged and resolved?

Based on the outcome, update your documented procedures, communications, and configuration settings. Regularly reviewing real-world incidents ensures your continuity plan remains accurate, actionable, and aligned with your organization's evolving needs.

13. Frequently Asked Questions (FAQ)

Below are some common questions related to Rublon MFA outages and business continuity, along with answers:

Q1: How can I tell if a connector has entered “Fail Mode” (i.e., is bypassing MFA or denying access due to the Rublon API being unreachable)?

A: There are a few indicators:

If you have set a connector to **Fail Safe** (FailMode=bypass), the obvious sign is that users are logging in **without being prompted for MFA** when they normally would be.

For example, if suddenly no one is getting prompted for MFA on login, it’s likely the connector failed open (bypassing) – either due to an outage or a config issue. On the other hand, if **Fail Secure** (FailMode=deny) is in effect, users will be complaining that they cannot log in at all.

To confirm Fail Mode activation on the backend, check the connector’s configuration. You can also check the connector’s log file, as many Rublon connectors will log an event when the Fail Mode triggers.

Currently, Rublon API does **not** send a direct alert when a local connector goes into Fail Mode because it may not even know it if the connector cannot reach the Rublon API. We recommend you use a monitoring system or SIEM to watch the logs of your critical connectors. For example, you could set up an alert if there is a sudden spike in “bypass” authentications. Proactive monitoring can give you an early warning that something’s wrong.

Q2: Will Rublon MFA work if there is no internet access or the user is completely offline?

A: Generally, no. Rublon MFA is a cloud service, so the client must contact the Rublon servers during login. If there is no internet connectivity, the Rublon MFA step cannot be completed, and Fail Mode logic applies (bypass or deny). The one notable exception is Rublon for Windows’ Offline Mode feature.

14. Appendix A: Glossary of Key Terms

- **Rublon API:** The central web service that handles multi-factor authentication requests.
- **Rublon Prompt:** The interactive screen displayed after successful primary authentication in GUI-supported Rublon-integrated applications. It allows users to select an available authentication method and enroll new devices for multi-factor authentication.
- **Primary Authentication:** The initial process of verifying a user's identity when they first attempt to access a system or application, typically involving credentials like username and password.
- **Secondary Authentication:** The process that provides an additional layer of verification after primary authentication is successful, typically involving second-factor authentication methods (e.g., TOTP codes, push notifications, or FIDO security keys) to further confirm the user's identity.
- **Second Factor:** A verification method used as part of multi-factor authentication, distinct from a user's primary credentials (username + password). It could be something the user has (e.g., a mobile phone) or something they are (e.g., a fingerprint).
- **Fail Mode:** A setting determining how an application behaves if it cannot contact the Rublon API. Fail Mode can be set to Fail Safe (bypass user without MFA) or Fail Secure (deny user access).
- **Fail Safe (Bypass):** Allows a user who passes primary authentication to log in without MFA if the Rublon API is unreachable.
- **Fail Secure (Deny):** Denies login entirely if MFA cannot be completed.
- **Bypass Code:** A one-time code that administrators can generate to let a user log in without completing an MFA prompt (e.g., if the user has lost access to their second-factor device).
- **Service Degradation:** A partial outage where the Rublon API is still reachable, but one or more MFA services (Admin Console, SMS Delivery, etc.) may be down, slow, or unreliable.

Rublon sp. z o.o.
ul. Stanislawa Wyspianskiego 11
65-036 Zielona Gora
Poland

www.rublon.com