

Systemic Weaknesses in Identity-Centric Security Models

WHITE PAPER

Author: Samuel K. Thomas - Founder & CEO of QILAHK | Cybersecurity Trust Authority

Security leaders and architects
October 10, 2026

Table of contents

Abstract	3
Executive summary	4
Decisions for identity and security leaders	4
Identity as a shared security dependency	5
Identity security and resource protection	5
Mapping dependencies behind identity authority	6
Enrollment and recovery weaknesses	6
Federation and the interpretation of assurance	7
Central authority and administrative concentration	8
Machine identities and automated delegation	9
Contextual signals and circular trust	9
A bounded incident perspective	10
Strong authentication within the full dependency chain	11
Assigning responsibility across the identity lifecycle	13
A 90 day resilience program	13
Recovery as an assurance decision	14
Containing administration and machine authority	14
Exercising failure and restoration	15
Measures for leadership review	15
Action checklist	16
Recommended next step	16
References	17

Abstract

Identity-centric security places identity evidence and relationships at the center of access decisions, creating dependencies that extend beyond the primary login mechanism. This white paper examines systemic weaknesses in enrollment, recovery, federation, administration, machine identity, contextual signals, and authority withdrawal. It distinguishes authentication assurance from resource entitlement and models how shared authorities can affect several downstream services. Published standards and a bounded public incident account support analysis of dependency concentration, circular trust, and containment. The proposed approach maps authority-producing relationships and evaluates which independent controls continue to protect sensitive operations when a selected dependency becomes unreliable. Operational recommendations assign lifecycle responsibilities, constrain consequential administrative changes, preserve delegation context, and test recovery and restoration through a 90-day resilience program. Strong authentication remains valuable, while its contribution must be assessed alongside the surrounding dependency chain. Hypothetical scenarios illustrate the framework; the paper does not claim a complete enterprise model, measured resilience improvement, or newly conducted experiments.

Executive summary

Identity-centric security works best when identity evidence is combined with resource-level authorization and independent limits on authority. Strong authentication protects an important part of the system, but enrollment, recovery, federation, administration, machine identities, and contextual signals also determine who can act. A weakness in one shared dependency can affect several applications that each behave correctly with the evidence they receive.

Security leaders should map the dependencies behind sensitive operations and test containment under bounded failures. The key question is which decisions remain protected if one identity, recovery path, administrator, or signal producer becomes unreliable. Separate authentication assurance from entitlement, constrain administrative changes, and verify withdrawal of authority wherever it has been copied into sessions or delegated credentials.

This white paper provides a dependency framework, a bounded incident perspective, and a practical resilience program. It does not argue for abandoning centralized identity or claim that more independent systems always improve security. The recommended approach preserves centralized governance while limiting consequential authority paths. The immediate decision is to make recovery, administration, and machine identity part of the same resilience assessment as the primary login mechanism.

Decisions for identity and security leaders

Approve a shared distinction between identity assurance and resource authority. Identity services establish accounts and authentication evidence under specified conditions. Resource owners determine which operations those principals may perform. An enterprise account should not become a universal permission category. Applications should consume the assurance context they need and retain independent checks for organization membership, resource ownership, and sensitive transactions.

Identify the common dependencies behind critical services. Include federation issuers, privileged administration, recovery channels, directory membership, device management, and workload credential issuance. Separate products can still depend on the same administrator or recovery mechanism. Conversely, a centralized service can support strong containment when its administration and consumers are well constrained. Evaluate actual authority relationships rather than treating product count as a measure of resilience.

Prioritize weaknesses by the scope of authority they can influence. A recovery process that can replace credentials for privileged accounts deserves different scrutiny from one restricted to low-impact accounts. A workload identity shared across sensitive services can have broader consequences than an isolated runtime identity. Decisions should reflect the operations reachable under the defined failure scenario, including the independent controls that continue to constrain them.

Identity as a shared security dependency

Identity-centric security places identities and their relationships near the center of access decisions. Organizations increasingly need to distinguish remote users, third-party collaborators, automated workloads, and devices across environments that no longer share one stable network perimeter. An identity-based approach can support consistent policy across those environments. The difficulty is that identity assurance becomes an input relied upon by many services, so errors in its creation or interpretation can have broad consequences.

This white paper studies systemic weaknesses: conditions that affect several downstream decisions because those decisions share an authority, signal, process, or failure mode. A weak recovery process can undermine multiple applications that rely on the same account. An administrator who can alter both federation configuration and audit collection can influence acceptance and evidence together. A workload identity reused across services can connect otherwise separate data domains. These are dependency problems rather than merely isolated credential defects.

Identity security and resource protection

NIST's zero trust architecture rejects implicit trust based solely on network location or asset ownership and distinguishes authentication from authorization (Rose et al., 2020). That distinction is important: zero trust does not mean that an authentic identity is universally trusted. Resource decisions still require policy and relevant context. The identity system provides evidence used by the decision, rather than a universal exemption from further checks.

An identity-only interpretation narrows this architecture improperly. It might treat enrollment in the corporate directory as sufficient for every internal service, or treat an authenticated workload as entitled to all resources reachable through the service network. The result recreates a perimeter around the identity system. The architectural concern is the breadth of the inference, not the fact that identity has become a major decision input.

This white paper distinguishes four propositions that are often collapsed. A real person has been associated with an account. A claimant controls a recognized authenticator. An assertion has been validated by a consuming application. The resulting principal is authorized for a particular operation. Each proposition can depend on different evidence and different authorities. Assurance about one does not logically establish all the others.

NIST's identity guidance separates proofing and enrollment, authentication and authenticator management, and federation and assertions (National Institute of Standards and Technology [NIST], 2025a, 2025b, 2025c). This separation motivates the dependency model. An assessment of identity security should follow the transitions between those functions rather than treating the directory entry as the beginning and end of identity assurance.

Mapping dependencies behind identity authority

The model represents identity authority as a directed dependency graph. Nodes include enrollment processes, credential stores, authenticators, federation issuers, validation configuration, directories, administrative roles, contextual signal producers, and resource enforcement points. An edge means that one node relies on another to create, interpret, modify, or withdraw a security-relevant proposition. The graph is an analytical aid; it is not claimed to be a complete formal model of an enterprise.

A dependency matters when failure at its source can change a downstream decision. For example, a directory supplies group membership to a policy service, which then authorizes an administrative console. If the directory's membership update is incorrect, the console may behave exactly as configured while granting inappropriate access. The downstream enforcement mechanism is functioning, but its trusted input no longer represents approved authority.

The graph should record the type of authority carried by an edge. Authentication authority establishes a principal under a specified assurance context. Policy authority grants permissions. Administrative authority changes the rules or credentials. Recovery authority restores control when ordinary evidence is unavailable. Observational authority supplies signals used to judge risk. Combining these types in one label called trust conceals important differences in control and accountability.

Common dependencies deserve special attention. Two applications may appear independent because they run in different environments, yet share a federation signing authority, recovery team, or administrator workstation. A failure involving that shared dependency can affect both. Conversely, separate identity providers do not guarantee independence if they share the same recovery email or privileged management path. Resilience analysis should follow actual dependencies rather than product boundaries.

Containment is the ability of downstream controls to limit the consequences of an incorrect upstream claim. A resource may require membership approved by its own owner, impose transaction limits, or demand an independent approval for a sensitive change. These controls can constrain an authentic but inappropriate principal. The framework evaluates containment by asking which decisions remain protected after a selected dependency fails and which paths remain available for legitimate recovery.

Enrollment and recovery weaknesses

Enrollment establishes the relationship between an account and the identity or role it represents. Errors at this stage can persist through later successful authentication. A claimant may correctly demonstrate control of an authenticator attached to the wrong account or organizational role. Stronger authentication then reliably repeats the wrong association. The system needs assurance about account creation and entitlement assignment in addition to assurance about credential use.

NIST's identity-proofing guidance addresses proofing and enrollment as distinct parts of the identity lifecycle (NIST, 2025a). For access design, organizational authority may require additional evidence beyond personal identity. Proving that someone is a particular person does not prove that the person

is currently an employee, may administer a tenant, or may approve a financial transaction. Those relationships require their own authoritative sources.

Recovery is a second path into account control. NIST's authentication guidance recognizes several recovery methods and calls for risk analysis and documentation of alternative methods (NIST, 2025b, §4.2). The important systemic question is whether recovery preserves the assurance required by consumers or silently creates a lower-assurance route to the same authority. An attacker need not defeat the primary authenticator if another route can replace it.

Consider a hypothetical privileged account protected by a phishing-resistant authenticator. A support process allows the authenticator to be replaced after a request from an email address that can itself be recovered through a weaker process. The primary ceremony can remain sound while the account's effective control depends on that weaker chain. The scenario illustrates a dependency to assess; it does not assert that any particular recovery method is categorically unacceptable.

A defensible recovery design records what the recovered account may do immediately afterward. Sensitive operations might require renewed evidence, delayed activation, or independent review, depending on requirements. Notifications and audit records support detection and correction, but do not substitute for assurance at the decision point. Research should examine both unauthorized recovery resistance and the ability of legitimate users to regain access reliably.

Federation and the interpretation of assurance

Federation allows applications to rely on another authority's identity assertions. It can reduce duplicated credential handling and improve consistency, while increasing the importance of the relationship between issuer and consumer. A consumer must know what the assertion establishes, which identities the issuer is entitled to describe, and how the assertion maps into local authority. Without these distinctions, one federation connection can become a broadly reusable privilege source.

NIST's federation guidance requires trust agreements to define relevant terms, including acceptable assurance levels and responsibilities between participants (NIST, 2025c, §3.5). OpenID Connect defines issuer, audience, and related validation expectations for identity assertions (OpenID Foundation, 2023). These requirements show that federation is a governed relationship as well as a technical message exchange.

Assurance can be lost in translation. An application may receive information about authentication context and then create a local session that no longer records it. Later, the session may be treated as equivalent to a stronger or more recent authentication event. The system has converted a contextual claim into a categorical identity. The dependency model records that translation so that the consumer's assumptions remain inspectable.

Tenant boundaries are another interpretation concern. A trusted provider may authenticate users for several organizations, but the application must still establish which organization and resources each user may access. Shared federation infrastructure does not imply shared business authority. A local

membership decision should have a clear source and lifecycle, especially when invitations, mergers, or contractor arrangements change the relationship.

The recommended control is explicit assurance consumption. Resource owners specify the evidence required for operation classes, and applications preserve the context needed to evaluate those requirements. A successful assertion validation should produce a bounded result rather than an unrestricted label of trusted user. This proposal does not require every request to trigger a new login; it requires session and policy semantics to state when existing evidence remains sufficient.

Central authority and administrative concentration

Centralized identity can improve consistency, visibility, and operational efficiency. It can also concentrate the ability to change identities and access rules. The relevant risk is not centralization by itself, but the scope of changes available through one administrative path and the independence of controls that constrain that path. A well-designed centralized service may offer stronger containment than fragmented systems with weak governance.

An administrator may be able to reset credentials, add group members, register applications, alter federation settings, and change audit destinations. These abilities should be examined as distinct powers. If one actor can grant authority and suppress the evidence of granting it, downstream applications may have limited ability to distinguish approved changes from misuse. The model records both operational privileges and their monitoring dependencies.

Separation of duties can reduce this concentration when applied to meaningful changes. A second approval for a high-impact federation change is more relevant than a generic second login to the same administrative account. The independence of the second approver matters too. If both approval identities can be recovered through one compromised channel, the process has fewer independent barriers than its workflow diagram suggests.

Administrative workstations and automation credentials form part of the identity control plane. A secure issuer with an exposed management path still depends on the management path's integrity. Conversely, hardening the management path without constraining the actions available through it can leave excessive blast radius after compromise. The design should address both the likelihood of control-plane misuse and the scope of possible changes.

The governance principle is bounded administrative authority with accountable change. High-impact modifications should have an owner, justification, affected-consumer inventory, and a rollback or containment plan. Review should ask whether a compromised administrator could expand authority across unrelated domains and whether another component would notice or limit that expansion. This is a resilience question that remains relevant even when administrator authentication is strong.

Machine identities and automated delegation

Identity-centric discussions often focus on human login, but automated workloads can exercise extensive authority without a human ceremony. A machine identity represents an execution context and its permitted relationships. Its security depends on how that identity is issued, which workloads can obtain it, where credentials are exposed, and how authority is withdrawn when the workload changes or disappears.

Kubernetes service accounts provide identities for processes associated with workloads and have documented token and authorization relationships (Kubernetes, n.d.). The broader analytical concern is identity placement: an architecture must establish that the workload receiving a credential is the workload approved to use that authority. A valid service-account token identifies a principal, but does not prove that the executing code still matches its approved purpose.

Shared identities can obscure accountability. If several services use one credential, logs may identify the shared principal without identifying the initiating workload. Incident response must then reconstruct the source through additional evidence. Shared credentials may be intentional in a small environment, but the tradeoff should be explicit. Their authority and rotation scope can be broader than operators expect.

Delegation adds another dependency. A service may act for a user, a tenant, or another service. If the downstream request preserves only the service identity, the resource may lose the basis for applying the original subject's restrictions. If it preserves only a user-supplied subject field, it may mistake unverified annotation for authenticated provenance. The architecture needs a defined representation of actor, subject, and permitted delegation.

Workload lifecycle should therefore include issuance, renewal, placement, and retirement evidence. A deployment change can alter who receives an identity without editing the identity's permission policy. Review must connect those two layers. Verification should establish whether replacing a workload, changing its owner, or moving it between environments triggers the appropriate access review and whether residual credentials continue to permit operations beyond the approved interval.

Contextual signals and circular trust

Adaptive security uses signals such as device posture, location, behavior, and observed risk. These signals can improve decisions when their meaning and provenance are understood. They also create dependencies on collection agents, device enrollment, analytics services, and the identities used to administer them. A policy that consumes a signal must know what evidence produced the signal and how fresh that evidence is.

Circular trust appears when a system uses one claim to validate another that ultimately depends on the first. A device might be treated as trusted because an identity account enrolled it, while the identity account receives privileged recovery because it is accessed from that trusted device. This arrangement is not necessarily unsound, but its assurance cannot be counted twice. The analysis should identify the initial independent evidence and the conditions under which either relationship can change.

Signals can also be correlated. Device compliance, malware status, and certificate validity may all depend on the same management agent or administrative channel. Presenting them as three independent reasons for access can overstate resilience. The framework therefore records common producers and modification paths. More signals are useful when they add information, not merely when they increase the visible number of policy conditions.

A stale signal may remain formally well-structured while losing relevance. An earlier device assessment cannot establish the current integrity of an endpoint indefinitely. Systems must define acceptable age and failure behavior for each operation class. Blocking every request whenever telemetry is unavailable can impair essential work; accepting indefinitely old signals can undermine policy. The appropriate balance needs explicit requirements and operational testing.

Privacy is connected to this design. Identity-centric monitoring can aggregate behavior across services and reveal sensitive relationships. Collecting data without a clear decision purpose adds governance cost and can create another valuable target. This white paper recommends limiting signals to documented needs, controlling their consumers, and preserving sufficient decision evidence without making the identity system a repository of every possible user activity.

A bounded incident perspective

Microsoft's January 19, 2024 disclosure reported that Midnight Blizzard used password spraying against a legacy non-production test tenant account and then used that account's permissions to access some corporate email accounts. Microsoft stated that the incident was not the result of a vulnerability in its products or services (Microsoft Security Response Center [MSRC], 2024). These are the disclosure's reported facts, not an independent conclusion about every technical detail.

The incident is relevant because the description connects an initially compromised identity in one environment with authority affecting another. The architectural interpretation is that environment labels such as test or non-production should not be used as substitutes for evaluating effective access relationships. A low-sensitivity starting environment can still matter if its identities retain consequential authority elsewhere.

The disclosure does not establish the entire permission graph or prove that every recommended control would have prevented the incident. It cannot support a precise counterfactual claim that a particular architectural change would have eliminated the attack. It instead motivates a review question: which identities in less monitored or legacy environments can exercise authority over current sensitive resources, and who owns those relationships?

This question differs from asking whether an account belongs to the corporate directory. It examines the dependencies that make the account consequential. A complete review would identify credential protection, permitted actions, cross-environment relationships, administrative ownership, and termination conditions. Those details would need direct organizational evidence. The public statement provides a reason to investigate them, not a substitute for that investigation.

The example also preserves the distinction between product vulnerability and system weakness. A system can permit harmful use of valid permissions without an implementation defect in the identity

product. That observation does not remove accountability; it changes the likely remediation target. Configuration, legacy ownership, entitlement governance, and containment may require attention even when protocol libraries behave correctly.

Strong authentication within the full dependency chain

Strong authentication remains valuable in an identity-centric architecture. WebAuthn supplies relying-party and origin-related protections through its credential ceremonies (World Wide Web Consortium [W3C], 2021). OAuth security guidance addresses credential flow and token-use risks, while DPOP provides proof-of-possession mechanisms for relevant token use (Lodderstedt et al., 2025; Fett et al., 2023). These mechanisms address distinct parts of the dependency chain. Their presence should improve the assessment of those parts without being counted as evidence for unrelated enrollment, permission, or recovery properties.

Consider a hypothetical organization that replaces password-based privileged login with a phishing-resistant mechanism. The change strengthens the primary authentication path and should be evaluated for that benefit. The organization also maintains a recovery process, federation configuration, an administrative group, and application sessions. If those surrounding processes remain unchanged, the improvement does not automatically establish stronger assurance for every path into privileged authority. The correct analysis records both the improved mechanism and the dependencies that still need examination.

The first question is which account the authenticator controls. If enrollment attached the credential to an incorrectly provisioned account, the ceremony can authenticate that account reliably while the entitlement remains inappropriate. The second question is who can replace the authenticator or change the account's recovery data. The third is who can modify the resource policy that recognizes the account as privileged. These are separate control paths and should not be collapsed into one judgment about the authenticator's cryptographic strength.

The same principle applies to proof of possession. A token-use mechanism can require access to a proof key, reducing the usefulness of a credential copied without that key. It does not determine whether the token represents the right tenant or whether the resource grants excessive authority to its principal. The dependency model therefore associates the mechanism with the threat condition it addresses. If an endpoint or service retains both the credential and its usable proof capability, the assessment must consider that execution context rather than assuming the token is unusable to every actor except the original human.

Independent resource constraints can preserve containment when one identity path fails. A privileged configuration change may require an approval from a separate accountable role. A data export may be limited to the principal's approved organization and resource class. A deployment identity may be unable to select unrelated runtime identities. These controls need their own assurance and lifecycle analysis. They should not be praised merely for adding more steps; their value depends on whether they constrain a relevant authority path under the defined failure scenario.

Verification should compare the architecture before and after the authentication improvement using the same bounded scenarios. The expected improvement concerns outcomes for scenarios addressed

by the new mechanism, while checking whether recovery, administration, and session withdrawal remain unaffected. Such an evaluation avoids two misleading conclusions: that one strong mechanism makes the whole system secure, or that any remaining weakness makes the mechanism worthless. Security improvements can be real and consequential while remaining limited to the properties they establish.

This distinction also improves investment decisions. A team can prioritize the controls that reduce its most consequential dependencies rather than buying additional mechanisms with overlapping coverage. It can explain why a resource-level restriction remains necessary after an authentication upgrade and why a recovery redesign may deserve attention before another login feature. The argument is analytical, not a quantified return-on-investment claim. Actual prioritization requires evidence about the organization's workflows, exposure, and operational constraints.

Finally, the dependency chain must remain understandable to operators. An incident responder needs to know which mechanism established a session, which authority granted its permissions, and which action will withdraw them. If these relationships are hidden behind a single label of authenticated, operators may overestimate both protection and containment. Preserving meaningful assurance context and recording authority transitions support more precise response. This is the systemic complement to strong authentication: confidence in the mechanism should be accompanied by clarity about the limits of the authority derived from it.

An assurance claim should identify its consumer. A security team may describe an account as strongly authenticated, while an application owner needs to know whether the current session is appropriate for a particular operation. Translating between those perspectives requires a clear statement of what was verified and when. A label useful for identity administration can become misleading when reused as a universal permission condition. The dependency model retains the relationship between evidence and the specific decision that consumes it.

Assurance should also survive organizational change. An account can move between departments, an application can change owners, and a provider relationship can expand to new populations. Those changes may leave authentication mechanisms intact while altering the approved authority associated with the identity. Periodic review should focus on those relationships, especially where a legacy account or workload remains technically active without a current accountable owner. Historical enrollment establishes a past event; ongoing entitlement requires a continuing basis.

These observations suggest a practical responsibility model. Identity operators own the evidence their system supplies, resource owners own the permissions they grant, and platform operators own the reliability of the paths carrying those decisions. Incident response connects those responsibilities when authority must be withdrawn. Clear ownership cannot guarantee correct behavior, but it makes gaps easier to investigate and correct. Empirical evaluation should test whether this assignment improves response precision and reduces unresolved dependencies, rather than assuming that an organizational chart alone establishes effective coordination.

Assigning responsibility across the identity lifecycle

Enrollment owners should define the evidence needed to create an account and associate it with organizational roles. Authentication owners should define the mechanisms and session context they establish. Recovery owners should document alternative routes to account control and the authority available immediately afterward. Federation owners should maintain issuer relationships and acceptance configuration. These responsibilities overlap operationally, but their evidence requirements should remain distinct.

Resource owners should define what additional authorization and assurance sensitive operations require. Platform owners should preserve the authenticated context between components and operate invalidation reliably. Privileged-access owners should constrain the changes administrators can make. A coordinating security owner should ensure that no function assumes another has supplied evidence that the architecture does not actually establish.

Review ownership when applications, workloads, or departments change. An account can remain technically active after its purpose disappears. A replacement workload can inherit an identity approved for another use. A provider relationship can expand to new user populations. Each change can alter authority without changing the primary authenticator. Lifecycle responsibility should therefore follow the continuing business relationship, not only the credential inventory.

A 90 day resilience program

Days 1 through 30 should map dependencies for a small set of critical operations. Trace account creation, credential management, federation interpretation, permission assignment, administrative change, and authority withdrawal. Include support-assisted recovery and emergency access. Record where several services share the same authority or management path. Confirm the map with owners and mark relationships that remain unknown.

Days 31 through 60 should address the highest-impact shared dependencies. Review privileged recovery, federation administration, broad group management, workload credential placement, and the ability to alter audit collection. Introduce separation where it constrains a meaningful authority path. Preserve legitimate recovery and administration through controlled testing. Additional approvals should be independent for the chosen scenario, rather than relying on accounts recoverable through the same compromised channel.

Days 61 through 90 should exercise containment and recovery. Simulate bounded failures in controlled accounts and services. Determine which sensitive operations remain reachable and which controls hold. Test whether operators can withdraw existing authority and maintain essential legitimate workflows when the primary identity dependency is unavailable or suspected unreliable. Retire emergency grants and verify the resulting access state after the exercise.

Repeat this program by operation class or service group. A useful wave produces a dependency map, accountable remediation, and evidence of containment and restoration. Avoid broad resilience claims based on one successful primary-login test. The program should expose which scenarios were tested and which administrative or provider-internal dependencies remain outside the available evidence.

Recovery as an assurance decision

Treat recovery as a controlled authority transition. State which evidence permits recovery, who may approve an exception, what changes are made, and what restrictions apply immediately afterward. A recovered ordinary account and a recovered privileged account need not have the same operational policy. The requirements should reflect the authority each can exercise and the legitimate user's ability to complete the process.

Test the full recovery chain. A verified email message may depend on a mailbox whose own recovery follows another process. A trusted device may depend on enrollment through the same identity being recovered. These relationships can be acceptable, but they should not be counted as independent evidence without analysis. Document the initial evidence that anchors the chain and who can modify it.

Support teams need clear escalation rules and decision records. Notifications and logs help detect unauthorized changes, but the process should not rely on detection alone when a weak control path can directly produce sensitive authority. At the same time, unnecessary barriers can strand legitimate users. Track successful recovery, delays, erroneous denial, and exception use alongside security outcomes.

Containing administration and machine authority

Review administrative powers as separate capabilities. Resetting a credential, assigning a privileged role, registering an application, changing a federation trust, and altering audit collection have different consequences. Determine which combinations one actor can exercise and whether another component will constrain or observe the change. Strong administrator authentication reduces some risks while leaving the scope of administrative authority relevant.

For workload identities, connect issuance to approved execution context. Record which deployments can receive the identity, who can select or replace it, and which resources it can access. Shared identities need an explicit justification and enough provenance to distinguish initiating workloads. Rotation and retirement should follow the approved purpose, especially when a service is replaced or moves between environments.

Delegated requests should preserve the actor and represented subject when downstream policy needs both. A service credential can establish workload identity without establishing the initiating user's permission. User-supplied annotation should not substitute for authenticated provenance. Define the delegation rule and apply resource checks at the component that owns the protected action.

Exercising failure and restoration

Choose bounded scenarios before conducting an exercise. Examples include compromise of one ordinary workload, incorrect group membership, unavailable federation, unauthorized recovery, or an unreliable device signal. Specify what the actor or failure can control and what remains independent. An exercise that assumes total compromise from the outset cannot show which architectural constraints limit a realistic intermediate failure.

Use controlled identities and resources to observe which operations remain possible. Record the resource checks, transaction approvals, and restricted identities that constrain the path. Measure withdrawal of existing sessions and delegated credentials separately from prevention of new issuance. Verify essential business workflows during disruption, because complete denial can conceal failure to meet continuity requirements.

Restoration should include verification that emergency access has ended and ordinary trust relationships have been re-established. Operators need a documented management path if the primary identity service cannot be relied upon. That path should be narrowly scoped and observable. The exercise should expose whether the fallback depends entirely on the same service it is intended to recover.

Measures for leadership review

Track coverage of critical-operation dependency maps, privileged recovery paths reviewed, high-impact administrative changes independently controlled, and workloads with owned issuance and retirement rules. Report containment outcomes by scenario, including the sensitive operations still reachable. Include withdrawal delays and failures to maintain essential operation. Avoid reducing the program to a count of authentication factors deployed.

Measure the quality of alternate control paths. Record recovery completion, exceptions, delays, and the authority available after recovery. Track unowned legacy accounts and workload identities by consequence, not just quantity. Review whether contextual signals add distinct evidence or share one common producer. These measures support decisions about the next dependency to address.

Maintain clear evidence limits. Provider-internal controls may be unavailable for direct inspection, and some administrative relationships may remain uncertain. Record those gaps and the assurance obtained through documentation or contractual commitments. An incomplete map should not be presented as proof that two controls are independent. Revisit the assessment when material ownership, recovery, or federation relationships change.

Action checklist

- Separate authentication assurance from resource entitlement.
- Map enrollment, recovery, federation, administration, signals, and machine identities.
- Identify shared dependencies behind critical operations.
- Constrain consequential administrative changes and verify meaningful independence.
- Preserve actor and subject context for delegated work.
- Test withdrawal of existing authority and continuity during bounded failures.
- Verify restoration and removal of emergency access after exercises.

Recommended next step

Choose one sensitive operation and trace every path that can create or change the authority needed to perform it. Include its recovery and administrative paths. Run one bounded failure exercise and one restoration exercise using controlled resources. Use the results to assign owners, close consequential gaps, and define the next service group for review. This creates an evidence-based resilience program around identity rather than relying on the strength of the primary login alone.

References

- Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., & Waite, D. (2023). OAuth 2.0 demonstrating proof of possession (DPoP) (RFC 9449). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc9449.html>
- Kubernetes. (n.d.). Service accounts. Retrieved October 10, 2026, from <https://kubernetes.io/docs/concepts/security/service-accounts/>
- Lodderstedt, T., Bradley, J., Labunets, A., & Fett, D. (2025). Best current practice for OAuth 2.0 security (RFC 9700). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc9700.html>
- Microsoft Security Response Center. (2024, January 19). Microsoft actions following attack by nation state actor Midnight Blizzard. <https://www.microsoft.com/en-us/msrc/blog/2024/01/microsoft-actions-following-attack-by-nation-state-actor-midnight-blizzard>
- National Institute of Standards and Technology. (2025a). Digital identity guidelines: Identity proofing and enrollment (NIST SP 800-63A-4). <https://csrc.nist.gov/pubs/sp/800/63/A/4/final>
- National Institute of Standards and Technology. (2025b). Digital identity guidelines: Authentication and authenticator management (NIST SP 800-63B-4). <https://pages.nist.gov/800-63-4/sp800-63b.html>
- National Institute of Standards and Technology. (2025c). Digital identity guidelines: Federation and assertions (NIST SP 800-63C-4). <https://pages.nist.gov/800-63-4/sp800-63c.html>
- OpenID Foundation. (2023). OpenID Connect Core 1.0 incorporating errata set 2. https://openid.net/specs/openid-connect-core-1_0.html
- Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST SP 800-207). National Institute of Standards and Technology. <https://csrc.nist.gov/pubs/sp/800/207/final>
- World Wide Web Consortium. (2021). Web authentication: An API for accessing public key credentials - Level 2. <https://www.w3.org/TR/webauthn-2/>