Identity and Access Management Design Best Practices

VMware Cloud Foundation 9.1

June 25, 2026

vmware by Broadcom

Table of Contents

Disclaimer

This document is intended to provide general guidance for organizations that are considering Broadcom solutions. The information contained in this document is for educational and informational purposes only. This document is not intended to provide advice and is provided “AS IS.” Broadcom makes no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the information contained herein. Organizations should engage appropriate legal, business, technical, and audit expertise within their specific organization for review of requirements and effectiveness of implementations.

Executive Summary

A stolen or socially engineered identity is one of the most common ways into virtualization infrastructure, often reaching the management plane without a software exploit, and AI is making those identity attacks cheaper, faster, and more convincing. The most effective response is architectural: isolate the identity systems that infrastructure trusts, so that a compromise of corporate identity does not become a compromise of VMware Cloud Foundation (VCF). This is the identity piece of a larger defense; segmentation, logging, patching, and recovery are assumed to be pursued alongside it.

The key controls:

- Build a separate, infrastructure-only identity provider, with no trust relationship to corporate identity systems.
- Federate VCF management to that provider through VMware Cloud Foundation SSO (VCF SSO), and require phishing-resistant MFA from managed administrative devices.
- Keep systems that can change accounts, credentials, or authentication settings, and monitoring tools that can execute or push, off the infrastructure identity provider.
- Make a stolen credential insufficient on its own: public-key MFA, conditional access, device posture, and short-lived tokens, with session revocation where the identity provider and components support it.
- Drive host administration through VMware vCenter with the VMware ESX hosts in lockdown, and restrict every path into vCenter and VCF Operations to those who need it.
- Hold service accounts and connected systems to least privilege, and keep break-glass and account recovery out of the corporate help desk.
- Retain and monitor identity logs, watching successful logins, not only failures.
- Run backup and recovery in their own isolated environment, on a dedicated VCF deployment.

The appendices give the priority order, implementation path, validation tests, and operational runbooks. Each control, the attacks it counters, and its trade-offs are taken up in turn below.

Introduction

The first ransomware attack happened in 1989. It spread by floppy disk to medical researchers, scrambled file names, and demanded $189 sent to a post office box in Panama. The mechanics have changed; the business model has not. What changed is reach. Widespread connectivity, cryptocurrency, a mature criminal supply chain, and stolen credentials turned a novelty into an industry.

Attacks against virtualization have moved toward the highest-value targets. Early ransomware encrypted individual operating systems one at a time. Today's variants go after the management plane and the backups, because an attacker who controls vCenter, the ESX hosts, or the backup systems controls every workload at once and removes the victim's ability to recover. In practice this makes the VCF management plane Tier 0 infrastructure: compromise the identity that controls it, and the reachable blast radius is everything it hosts.

Attackers reach that management plane in more than one way. Some exploit exposed, unpatched interfaces directly; the ESXiArgs campaigns encrypted hosts that were reachable over the internet with an unpatched service, an argument for patching and for keeping management interfaces off the open network. Others simply log in, with valid credentials obtained earlier through password spraying, stolen sessions, or bypassed multi-factor authentication. Identity compromise is a major and recurring path into infrastructure, even if it is not the only one. IBM's X-Force found that abuse of valid accounts tied for the top initial-access vector in 2024, at about 30% of the cases it handled, alongside exploitation of public-facing applications [3]. Mandiant's 2026 report ranked voice phishing the second most common initial infection vector in its 2025 data [1]. Exploited vulnerabilities lead in some data sets, including Verizon's [2], a reminder that patching matters too. Identity is the thread these recommendations follow.

Read the published incident reports and a pattern repeats:

- Initial access through a compromised identity, often an ordinary corporate account
- Lateral movement toward infrastructure
- Privilege escalation inside a management interface
- Mass encryption or destruction, backups included

Identity is the thread that runs through all four steps. If a compromise of the corporate identity provider cannot reach the infrastructure identity provider, many of these paths break. The aim here is to isolate the identity systems that infrastructure trusts, so that losing a general-purpose corporate identity provider does not mean losing the data center with it.

How AI Changes the Threat

None of the attacks above are new. What is new is the cost and expertise required to run them. Machine assistance does not replace every skilled attacker function, but it lowers the cost of several:

- Reconnaissance and target profiling, drawn from public code, configuration guides, and design documentation.
- Social engineering, with phishing and vishing lures tailored and produced at volume.
- Credential triage, sorting a stolen trove for the accounts that hold privileged access.
- Malware modification and parts of exploit development.

Each of these used to take skill and time; now they take less of both.

Velocity is the first thing that changes. When an exploit can be developed and launched faster than a slow approval cycle can clear a defensive change, the cycle itself becomes the exposure, which is an argument for pre-authorizing emergency security changes and fast-tracking fixes for actively exploited vulnerabilities, not for abandoning change control. The other change is who can play. AI is a great equalizer: attacks that once demanded a skilled, well-funded team are now within reach of almost anyone with a GPU, which widens the pool of capable attackers well beyond the specialists who used to hold those skills. Openness compounds both, because published code, configuration guides, and architecture diagrams benefit defenders and attackers' tooling alike, while attackers do not reciprocate by sharing their own.

Patching, while essential, is one piece among many, and it cannot by itself keep pace. A clean record to date is not proof of a strong defense; it may only mean the test has not yet come, or that an intruder is present and unnoticed. Resilience comes from architecture: identity isolation, segmentation, least privilege, logging, and a tested path to recovery, layered so that the failure of any one control is contained rather than catastrophic.

Multiple industry groups have reached the same conclusions independently. The Cloud Security Alliance [5] and FS-ISAC, along with the vendors building these AI tools, converge on a short list: isolate identity and require phishing-resistant multi-factor authentication, segment networks so a foothold does not become free movement, and design for breach with hard barriers rather than soft friction. Identity sits at the top of that list, because it is the control an attacker most wants to subvert and the one that, done well, most effectively bounds the blast radius.

AI-Assisted Impersonation

Plan for the possibility that someone will, at some point, call, write, or video-call pretending to be one of your administrators, and that the impersonation will be convincing. Of everything AI changes for identity, this is the change that should reshape your controls.

It used to take talent and time to impersonate a specific person. Now a few seconds of audio, scraped from a conference talk, a podcast, a voicemail greeting, or a social clip, is enough to clone a voice well enough to fool a colleague on the phone. Public profiles supply the rest: an administrator's name, role, manager, coworkers, the projects they are on, and the vocabulary they use. The same tooling produces tailored emails in a person's writing style and chat messages that answer

follow-up questions in real time. Voice cloning and tailored written impersonation are already practical; live video impersonation is improving and should be assumed credible for high-value administrative workflows, even if it is not yet as reliable as cloned audio. An attacker no longer has to know the target. The attacker can manufacture a convincing version of them from what is already public.

The signals human verification quietly relies on stop being reliable. Caller ID can be spoofed. Knowledge-based questions fail when the answers, a date of birth, a manager's name, the last digits of an employee ID, are discoverable online. "It sounded like her" is no longer evidence, and "I saw him on the video call" is becoming less of one. The help desk, the approver waiting to authorize an exception, and the engineer asked to enroll a new MFA device are all being asked to make a trust decision about an identity using exactly the signals that synthetic impersonation reproduces best. The FBI and CISA have documented attackers talking support staff into resetting passwords and moving MFA to attacker-controlled devices [6], and the New York State Department of Financial Services has warned that AI-generated deepfakes and voice cloning are being used to defeat identity verification [7].

Training people to spot better fakes helps less as the fakes improve. What holds up instead is to take the human judgment call out of the loop for anything that touches privileged authentication, and to anchor verification in factors that impersonation cannot reproduce. A cryptographic authenticator bound to the real session does not care how convincing a voice is. An in-person proofing step cannot be performed by a cloned voice over the phone. A multi-party approval, where two authorized people independently confirm out-of-band, defeats the single convincing call. These are the controls the recommendations return to, and the friction they add is the point: it puts a barrier in front of the exact step an AI-assisted impersonator is trying to rush through.

From Help Desk to Hypervisor

In July 2025, Google's Threat Intelligence Group documented [4] how the group tracked as UNC3944, also called Scattered Spider, moves from a phone call to encrypted datastores. It is worth walking through, because it shows how the pieces connect and because none of it is hypothetical. The chain is identity from end to end, and it runs in hours.

It starts at the help desk. The attacker calls posing as an employee and talks an agent into resetting a password or enrolling a new MFA device, using public information and, often, a cloned voice to pass verification. Inside the corporate directory, the attacker reads internal wikis and the directory itself for the groups that grant virtualization access, the ones helpfully named for the job, then calls back as a member of one. With that access they log into vCenter, reboot the vCenter appliance into a single-user boot mode for a passwordless root shell on the appliance, then enable SSH on the ESX hosts and reset their root passwords.

From there they detach the virtual disk of a powered-off domain controller to copy its identity database offline, add themselves to the backup administrators' group so recovery is gone too, and finally push ransomware to the hosts and encrypt the datastores. Because much of this uses valid credentials and low-level host operations rather than malware, and because endpoint detection has historically had limited visibility into ESX and the vCenter appliance, little of it trips an alert.

Trace the same chain against the controls here and several of its links break. The opening move is much harder if the help desk cannot reset an infrastructure administrator's authenticator and privileged resets require in-person or multi-party verification. The pivot from corporate directory to vCenter does not work if infrastructure identity is isolated, so corporate group membership grants nothing on the management plane. The recon step is slower if the privileged groups are not obvious. Direct host access is sharply curtailed if ESX runs in lockdown with SSH and the local shell off, forcing administration through vCenter's monitored path rather than a direct host login. The backup sabotage fails if backup identity is isolated from the directory the attacker just took. And whatever does happen is visible if the right logs are retained and watched. These controls do not make compromise impossible; a vCenter vulnerability, a stolen token, a malicious administrator, or a compromised isolated provider can still do damage. What they do is remove the easy, common links in this chain and force an attacker onto noisier, harder, or separate paths.

This is the whole argument in miniature: bound the blast radius so that an initial compromise does not become a complete one.

Understanding and Evaluating Trust

Every authentication and every authorization decision is an implicit trust relationship, and every one of them is something an attacker can try to use. Trust is also transitive: if system A trusts system B, and system B trusts system C, then A trusts C whether anyone intended it or not.

Mapping that trust is the first piece of work. The obvious relationships are easy to see: federation between identity providers, directory integrations, named service accounts. The dangerous ones are quieter: a shared service account, two management systems with overlapping scopes, a handful of tools that all authenticate against the same source. These relationships accumulate as an environment grows, and they create paths that no one designed and no one is watching.

Consider the common case where one corporate identity provider authenticates users to both office applications and infrastructure management. That single decision links the corporate identity system to the data center. Compromise the corporate identity provider and you have a path to infrastructure, which is exactly how a number of breaches have unfolded: credential theft in the corporate environment, then a short walk to vCenter. The same logic applies to the management tools that are trusted to hold privileged access to identity providers, such as monitoring, provisioning, and configuration management. Each is another path in, and each is worth questioning.

A trust evaluation examines authentication flows, authorization decisions, credential storage, and access patterns, and it asks a blunt question of every relationship: is this necessary, and could the same business outcome be reached with more isolation? Document the relationships, review them on a schedule, and justify each one. A trust relationship that cannot be justified is a finding, not a feature.

Zero Trust Architecture

Zero trust architecture is a shift away from perimeter-based security, and it is more than adding multi-factor authentication. MFA is often the headline, but treating zero trust as "we turned on MFA" misses the point. At its core, zero trust reduces the number of components that must be trusted implicitly and replaces assumed trust with explicit verification for every interaction. That principle is uncomfortable in traditional infrastructure, which quietly trusts a great deal: network switches, server firmware, storage arrays, management agents.

Zero trust can simplify the trust graph by removing implicit trust, though the implementation may add operational complexity of its own: new brokers, device-posture checks, policy and logging integrations, and exception workflows. The goal is fewer unexamined trust paths, not fewer moving parts in every case. Perimeter models lean on long chains of implicit trust, and cloud identity deployments in particular can spread an authentication path across several services and trust boundaries. Requiring explicit verification, and removing the trust that is not needed, leaves a smaller set of relationships you can actually reason about. Most of the recommendations here are subtractive: remove a federation, remove a connected system, remove a standing permission.

Scope

This guidance is focused, not exhaustive. It is about the identity and access architecture around VCF 9.1: which identity providers the infrastructure trusts, how authentication reaches the management plane, where authorization decisions are made, and how the same isolation extends to the backup and recovery environment an organization relies on to survive an attack.

It is not a configuration baseline or an exhaustive hardening guide. It makes architecture-level identity and access recommendations and points to the Security Configuration and Hardening Guides for exact settings, ciphers, and values. It is not a complete defense against AI-accelerated attacks, which also requires segmentation, logging, patching, and recovery; it is the identity piece of that larger effort, and it assumes the others are being pursued in parallel.

Some of what follows runs against long-standing enterprise habit. The instinct to connect every system to one identity provider, for one sign-on and one place to manage accounts, is strong and, in ordinary times, reasonable. The argument here is that infrastructure is not an ordinary case, and that the convenience of universal single sign-on is exactly the property an

attacker monetizes. Where a recommendation will be unpopular, the reason is stated alongside it, so you can weigh the trade-off for your own environment rather than take it on faith.

Intended Audience

This guidance is for organizations that have deployed, or are considering, VCF 9.1. Much of it applies to earlier vSphere and VCF releases as well, but the product names and capabilities described here, such as VCF SSO and the VCF Identity Broker, follow the VCF 9.1 generation.

If you consume VCF through an engineered system or a hybrid cloud product that packages VMware infrastructure as part of its own solution, check with that product's support before making changes based on this document. Broadcom cannot speak for third-party solutions built on top of the platform.

Key Terms

The recommendations turn on a few distinctions, so this guidance uses these terms precisely:

- **Corporate identity provider**: the general-purpose directory or cloud identity tenant that authenticates the workforce for email, collaboration, and business applications.
- **Infrastructure identity provider** (isolated IdP): a separate provider that authenticates only infrastructure administration, with no federation, synchronization, or shared administration with corporate identity, and whose own dependencies are kept off the corporate side.
- **Federation**: a standards-based trust, using SAML or OIDC, that lets one system accept authentication performed by an identity provider.
- **Synchronization**: automated copying of accounts, groups, or attributes from one directory into another.
- **Shared administration**: the same people or systems holding administrative rights over both providers.
- **VCF management plane**: the components an administrator controls the environment through, including vCenter, VCF Operations, VMware NSX, and VCF Automation.
- **Infrastructure administrator** (privileged infrastructure account): a person or account with rights to change the management plane or the hosts beneath it.
- **Service account / workload identity**: a non-human identity used by automation or an integration; a workload identity is purpose-built for that role rather than a repurposed user account.
- **Break-glass account**: an emergency account that does not depend on normal authentication paths, used when those paths are down or compromised.
- **Management network / controlled path**: a restricted network for management interfaces, reached only through a deliberate route such as a bastion or a zero-trust access broker rather than directly.
- **Recovery environment**: the backup and recovery systems an organization relies on to restore after an incident.

Security Hardening and Design

This is not an exhaustive configuration baseline. It makes architecture-level recommendations and leaves the exact settings, ciphers, and values to the Security Configuration and Hardening Guides at https://github.com/vmware/vcf-security-and-compliance-guidelines [15].

Isolate Infrastructure from Enterprise Identity Providers

Build a separate identity provider for infrastructure and give it no trust relationship to your corporate identity systems.

Counters: corporate identity-provider compromise, the Active Directory-to-hypervisor pivot, credential theft.

Most organizations authenticate everything, including infrastructure management, against one corporate Active Directory or cloud identity tenant. It is convenient, and it means that a single compromise of that one system reaches the infrastructure too. The corporate identity provider is also the most exposed identity system you run: it authenticates laptops, email, and the help desk, which is precisely where phishing, password spraying, and social engineering land first. Tying the data center to it inherits all of that exposure.

An alternative is an identity provider that exists only for infrastructure, with no federation, synchronization, or shared administration with the corporate side [14]. The initial build is project work; the recurring cost is real but lower than operating a general-purpose enterprise identity provider, and it should be budgeted for: patching, high availability, backups, monitoring, access reviews, credential recovery, disaster-recovery testing, and audit evidence. The payoff is a boundary: if corporate identity is compromised, the infrastructure identity provider is not on the other side of that compromise.

An identity provider that serves only infrastructure serves a small, well-understood population, so its access policies stay short enough to read and audit. A corporate identity provider that tries to be everything to everyone accumulates conditional-access rules, exceptions, legacy protocols kept alive for one old application, and integrations no one remembers approving, and attackers make their living in the gaps between those rules. Fewer rules mean fewer gaps.

The smaller population also generates far less authentication traffic, so the logs are smaller and cheaper to keep, which turns multi-year retention of infrastructure identity logs into an affordable line item rather than a budget fight. The same small, security-aware population lets you hold infrastructure identities to a higher standard than a general workforce would accept: longer passphrases, screening against breached-password lists, and phishing-resistant MFA applied without exception. VCF administrators can affect the availability, integrity, and operational control of large numbers of hosted workloads and often function as Tier 0 administrators, and they understand that level of access comes with accepting more friction to protect it.

The isolation has several parts.

Dedicated Infrastructure Identity Provider

Stand up an identity provider used only for infrastructure access. This can be a dedicated Active Directory forest, a separate cloud identity tenant, or another identity solution. It should have no trust relationship or federation with corporate identity systems, and no connection to the user-provisioning or system-management tools that reach across the organization. Every such relationship is trust, and trust is what an attacker reuses to cross from one domain into the next.

Separate and Distinct Administrative Accounts

Give infrastructure administrators dedicated accounts, unconnected to their corporate credentials. These accounts should live under their own password policy, their own lifecycle, and their own controls. An administrator's corporate account being phished should not hand over an infrastructure login as well. Better still, do not leave high privilege standing at all: grant it just in time, for the task and the window that need it, and check the most powerful credentials out of a vault that records the session. Privilege that exists only when needed is privilege an attacker usually finds switched off.

Independent Multi-Factor Authentication

Keep the infrastructure MFA control plane separate from corporate MFA. What matters is less whether an administrator carries a second authenticator than whether the same service governs enrollment, policy, recovery, and sessions for both. If one MFA service administers both, then MFA fatigue, a stolen token, a weak recovery path, or social engineering against that one service reaches both. Separate the enrollment authority, the recovery process, and the policy and session infrastructure, so the technique that defeats corporate MFA does not also defeat infrastructure MFA. A single hardware FIDO2 key can hold separate, origin-bound credentials for each side without bridging them; the boundary is the control plane, not the device.

Break-Glass Procedures

Provide an emergency access path that does not depend on corporate identity systems. When corporate identity is down or compromised, administrators still need a way in to respond. Break-glass credentials make that possible, and where you store them is itself a security decision: they must be reachable by the right people under pressure and out of reach the rest of the time. Treat their storage location with the same care as the accounts themselves. An emergency path you have never exercised is a guess; rehearse it so that "break glass" is a procedure and not an improvisation.

Regular Trust Validation

Watch for trust relationships that quietly re-bridge the boundary you built. Federation settings drift, a service account is granted a little more scope, a new integration goes in without review, and the boundary slowly stops being one. Audit federation, service-account permissions, and authentication flows on a schedule, and treat any new path across the boundary as a change that has to be justified.

Strict Account Recovery Processes

Keep infrastructure account recovery out of the corporate help desk. Staff who reset corporate passwords all day should not be able to reset infrastructure credentials. Infrastructure recovery should require multiple approvers, out-of-band verification, detailed logging, and, for privileged accounts, in-person identity confirmation. In-person reset is deliberately inconvenient, and the inconvenience is bounded: privileged infrastructure accounts should be few, so the slow path applies to a small population while covering the accounts that matter most.

This closes a path that has become a common threat-actor tactic: calling the help desk and talking an employee through a "password reset" for a privileged account. It has played out the same way across multiple intrusions, and it needs no exploit, only a convincing story and a help desk following its normal corporate procedure. The process works exactly as designed. It is the wrong process for that account. AI sharpens this further: cloned voices and tailored pretexts make the impersonation more convincing and cheaper to produce at scale, which is why a phone call or a support ticket is no longer adequate assurance for resetting a privileged authenticator.

Authenticator enrollment belongs in the same category: if a new MFA device can be registered with a password alone, the second factor protects nothing. Issue administrators more than one strong authenticator and keep a spare in secure storage, so a lost device is a routine swap rather than a reset request that has to be talked through a support channel. The same standard applies when a managed service provider or vendor operates the environment: their staff and tooling are a trust relationship like any other, to be scoped, logged, and bound by contract to the same reset and verification rules as your own team.

None of this is easy, because it cuts against the one-size-fits-all model that most identity programs are built to deliver. The point is not that universal single sign-on is wrong in general. It is that infrastructure is the wrong place to apply it, given how often a breached identity provider is the first domino in a ransomware attack.

Control Internet Exposure

You cannot fully isolate infrastructure from the internet, so reduce and control the exposure instead.

Counters: direct internet exposure of management interfaces and identity providers.

Few enterprise VCF environments are truly air-gapped, though some regulated, defense, and disconnected-site deployments genuinely run offline or nearly so. For the rest, patches, license checks, support tooling, and remote administration assume connectivity, and a cloud-hosted identity provider is internet-facing by design. That is a legitimate choice rather than a failure: an external identity provider reached over the internet is built that way from the start, with the internet as its transport layer. The mistake is not the connectivity itself, it is leaving it unexamined and treating a system as isolated when its real exposure has never been mapped. Where a cloud identity provider is in the design, its exposure is part of the threat model, which is one more reason to enforce phishing-resistant MFA and conditional access on it rather than leaning on network position for safety.

Treat inbound and outbound separately, because they carry very different risks. Direct inbound access to vCenter or ESX from the internet or a general-purpose corporate network should not exist; reach the management plane through a controlled path

such as a bastion or a ZTNA broker instead. Outbound paths, patch and update repositories, license checks, certificate validation, telemetry and support bundles, cloud identity-provider authentication, and vendor support tunnels, are often necessary, but each should be deliberate: outbound-only where possible, scoped to known destinations, and logged. The discipline that makes this real is a path inventory, listing every path that crosses the boundary with its owner, purpose, protocol, source, destination, authentication method, log location, and last review date. The honest version of this control is not "no internet." It is "no unaccounted-for internet," with every remaining path named, justified, and monitored.

Limit Administration Scope of Infrastructure Identity Providers

Keep anything that can change accounts, credentials, or authentication settings away from the infrastructure identity provider.

Counters: blast-radius expansion through connected systems, monitoring and remote-management abuse, identity-provider misconfiguration.

Isolating the identity provider is only half the work; the other half is limiting what is allowed to administer it. Anything that can create a user, reset a password, change a group membership, or turn off MFA is an effective administrator of everything that trusts the identity provider, and it can usually erase the record of having done so. The most dangerous actor in an identity system is often a legitimate one, which is why separation of duties matters: the ability to grant infrastructure access should not sit with the same people or systems that use it. That list is longer than it first appears: the obvious directory administrators, and the systems that feed the directory, HR-driven provisioning, automated password-reset tooling, configuration management, and any platform with write access to identity data. Each is a path to privilege that does not look like an administrator login in the logs.

There is a compliance consequence as well: because those people and systems can hand out or revoke infrastructure access, frameworks such as PCI DSS and NIST 800-171 bring an attached identity provider, its administrators, and its feeder systems into audit scope. Isolation does not remove that scope, but it shrinks it from the whole corporate directory to a small, dedicated team.

Do not connect systems that can alter accounts, passwords, or authentication configuration to the infrastructure identity provider or to infrastructure components. Provisioning systems, HR feeds, automated password-reset services, and configuration management belong on the corporate side of the boundary, not reaching across it.

Monitoring needs the same care, with a finer line. Telemetry collection should be read-only, because a path that only observes is hard to turn into a weapon. Remediation and automation are legitimate operating models, but they are not telemetry and should not ride the same broad, standing access; where a monitoring, EDR, SOAR, or VCF Operations workflow does act on infrastructure, route that action through a narrowly scoped service identity, an approved runbook, just-in-time authorization, a change record, and session logging, with an explicit limit on its blast radius. The pattern to avoid is a single monitoring integration that quietly holds standing authority to run anything, anywhere it watches.

Audit the infrastructure identity provider against published best practices [12] for whatever product you run, and revisit the audit as the product and the guidance change. Where the identity provider is a cloud service, pay attention to convenience features that widen the attack surface. On Microsoft Entra ID, for example, two convenience features deserve scrutiny for privileged accounts [13]. Avoid Seamless SSO for privileged infrastructure identities unless there is a specific, documented reason to use it; it reduces reauthentication friction and extends workstation and domain trust into cloud authentication flows, and if Conditional Access is misconfigured, that can weaken the separation you want between a compromised device and a privileged sign-in. And disable Self-Service Password Reset for privileged infrastructure administrators, or restrict it to phishing-resistant, administrator-approved recovery paths; do not let SMS, voice, email-only, or knowledge-based methods recover a privileged infrastructure identity. These features remove friction from everyday corporate use, and that is friction you want present on infrastructure identities.

Use Identity Federation to Introduce Multi-Factor Authentication

Use VCF SSO to federate the management plane to your isolated infrastructure identity provider, and use that federation to require strong MFA.

Counters: native MFA gaps on the management plane, credential-only logins.

VCF SSO, through the VCF Identity Broker, gives the VCF management components one identity configuration [18]. It configures vCenter, NSX, VCF Operations, and VCF Automation against a single broker, supports SAML, OIDC, and Active Directory over LDAPS identity providers, and is available in embedded, standalone, and clustered appliance deployment models. Pointing that broker at the isolated infrastructure identity provider gives you one place to enforce authentication policy for the management plane. VCF Automation extends this with a multitenant model in which different tenants can federate to different identity providers, which fits its role as the cloud interface for consuming VCF resources rather than for protecting the infrastructure management plane itself.

Individual VCF management components have historically offered limited or inconsistent multi-factor options of their own, so federating the management plane to an identity provider that enforces MFA is the practical way to apply one consistent, modern, phishing-resistant policy across all of them at once. Federation does not secure sessions by itself, though; SAML and OIDC have to be configured defensively, with signed assertions or tokens, strict audience validation, short token lifetimes, replay protections, secure redirect URIs, session revocation, and monitoring for impossible travel or anomalous token use. Configured that way, centralizing authentication lets you apply policies to infrastructure that would be impractical to mandate across an entire corporate user base. Because the infrastructure identity provider serves a small, privileged population, you can be stricter than corporate policy would tolerate:

- Require phishing-resistant FIDO2 keys for infrastructure access while corporate applications keep more flexible options
- Enforce shorter session timeouts on management access
- Apply stricter device-posture checks before granting access
- Restrict access from geographies your administrators do not work from

Federation adds a trust relationship and a dependency, which is the opposite of the subtractive direction the rest of these recommendations argue for, and that cost is worth stating plainly. The trade is deliberate: a single, well-controlled federation to an isolated identity provider buys standards-based authentication, centralized MFA, and one consistent policy across the management plane, and that is worth more than the one dependency it introduces. The danger is federating to the wrong thing. Federating the management plane to the corporate identity provider would hand an attacker the whole point of the exercise.

Make Stolen Credentials Insufficient

Assume a credential can be stolen, and make a stolen credential insufficient on its own.

Counters: phishing and vishing, push bombing, SIM swap, adversary-in-the-middle and token theft, infostealers.

Most of these attacks succeed by logging in, not by breaking in, and infostealer malware has turned credential theft into a commodity. The defensive corollary is to make a valid username and password, by themselves, not enough to reach infrastructure. Several controls stack toward that goal, and federating to an isolated identity provider, as above, is what makes them practical to enforce.

Start with the strongest authenticator you can deploy. Phishing-resistant means the authenticator is cryptographically bound to the legitimate site and cannot be replayed to an impostor: not SMS, voice, email codes, ordinary one-time codes, or push approvals, but FIDO2/WebAuthn keys, passkeys, or smart cards. Because the key is origin-bound, a credential phished through a proxy or an adversary-in-the-middle page is tied to the wrong origin and does not work against the real service [10].

They are not all equal, though. For the highest-value identities, VCF administrators, identity-provider administrators, break-glass custodians, and anyone who can reset an administrator's authenticator, prefer hardware-bound authenticators whose private key is non-exportable and held in a hardware-protected environment, such as FIDO2 security keys or smart cards, which is what NIST's highest assurance level requires; attestation is useful for verifying those authenticator properties but is not itself the requirement [10]. Syncable passkeys are a real improvement over passwords and one-time codes and may be acceptable for broader administrative populations, but because their keys can roam between devices they do not meet that bar and should not be the only factor guarding the most sensitive accounts.

One-time codes and push approvals are weaker still: a code can be relayed to an impostor site, and a push can be approved under fatigue. Number matching and added sign-in context are worth enabling, but they are hardening for push-based MFA, not the destination [11]. Pair whatever you use with conditional access, so authentication is judged against context and not the credential alone: where the request comes from, what device it comes from, and whether that fits the administrator's normal pattern.

Tie that context to the device. Enforce device posture and endpoint protection health as a precondition for access, so that an unmanaged or unhealthy machine cannot reach the management plane even with the right credentials. Stronger still, bind administrative access to managed devices with attested hardware identity, rooted in a TPM or a device certificate, so the question is not only "do you know the password" but "are you on a device we issued and can vouch for." The strongest form of this is a dedicated privileged-access workstation: a hardened, managed device used only for administration, with attestation, endpoint protection, and a locked-down browser, kept separate from the laptop used for email and the web. Privileged access to VCF should come from a managed, attested, healthy admin device, not from an unmanaged personal one.

Shorten the useful life of anything an attacker might capture. Lower authentication token lifetimes so that a stolen token expires before it is worth much, and keep a way to cut a session that has already started, because adversary-in-the-middle kits steal the session cookie after MFA succeeds, not the password before it. Sign-in frequency limits and continuous access evaluation let the identity provider re-check and revoke access when risk changes, instead of honoring a stolen session until it happens to expire. Replace long-lived secrets and shared service-account passwords with short-lived, narrowly-scoped tokens, which limits both the window and the blast radius if one is exposed. A credential valid only for minutes, only from an attested device, and only for a narrow scope is of little use to whoever captures it.

A valid credential is not the only thing worth stealing; a valid session is the other. Adversary-in-the-middle kits capture the session cookie or token after MFA has already succeeded, so password and MFA strength alone do not close the gap. Bind tokens to the device where the platform supports it, and keep token lifetimes and sign-in frequency short so a captured session is useful for less time. Where the identity provider offers continuous access evaluation, use it to signal revocation when risk changes; how quickly that takes effect depends on how soon each component re-validates the token. Keep administration off the browsers and workstations that also handle email and the web, where session-stealing malware lives. Put an additional barrier in front of destructive actions, through step-up reauthentication where the identity provider and component support it, or through a compensating workflow such as PAM approval, dual control, or a bastion, so a stolen session cannot, by itself, delete a datastore or a backup.

Severely Restrict Access to ESX and Appliance Management Interfaces

Keep direct access to ESX hosts and management appliances closed by default, and drive administration through vCenter and its RBAC model.

Counters: direct host compromise, lateral movement, controls and logging bypass.

The ESX host management interface, and the management interfaces of the various VCF appliances, are powerful and low-level. Direct access to them sidesteps vCenter's role-based access control and its logging, which are the controls the previous sections went to some trouble to build. Routing administration through vCenter means operations are subject to RBAC, are recorded in one place, and can be reasoned about; going straight to the host means none of that.

In practice this means lockdown for the hosts and a closed door for the appliances. ESX should run in lockdown mode with the local shell and SSH disabled, so that the host is administered through vCenter rather than logged into directly. Direct host access becomes a deliberate, logged exception for break-glass and specific support cases, not a standing convenience. Lockdown reduces the bypass paths; it does not make the host untouchable. Keep the exception-user list short, because adding routine ESX administrators to it defeats the purpose, and remember that whoever controls vCenter can still drive dangerous operations on the hosts it manages. The same logic applies to appliance management consoles: reachable through controlled paths by the few people who need them, and closed to everyone else.

There is a narrow benefit worth naming. When host and appliance interfaces are not directly reachable, a newly disclosed vulnerability in one of them is less directly exploitable from untrusted networks, because the attacker has to reach the interface before the vulnerability matters. Tight access does not patch the bug, and it does not lower the priority of patching Tier 0 infrastructure, but it can buy time to patch on your schedule rather than the attacker's.

Restrict Access to vCenter and VCF Operations

Limit access to vCenter and VCF Operations to people whose current job requires it, account for every path that reaches them.

Counters: management-plane takeover, reconnaissance, lateral movement.

Access to the vSphere Client through vCenter, and to the management functions in VCF Operations, should be limited to those who genuinely need it, and "need" should be read strictly. This is least privilege in practice: give each person, and each service, the minimum rights needed for the task, for the minimum time, and no more. Even broad read-only access is not harmless: a read-only view of the management plane is the reconnaissance an attacker wants, the inventory of what exists, what it is named, and where the high-value targets are, and AI tooling turns that inventory into a target list quickly. This is not only about the number of administrators. It is about every route into these interfaces, because a control on the front door is worth little if the side doors are open. The routes include:

- Direct administrative console access
- PowerCLI and API access
- Automation tool service accounts
- Backup system access
- Monitoring system access
- Third-party plugin access

Each is attack surface, and each should be minimized or removed. Most vSphere Client plugins, as an example, are not needed to run vSphere or VCF. Installing them creates dependencies and cross-connections that cut against separation of duties and least privilege, and they add friction to patching and upgrades, which slows your response when a vulnerability is disclosed.

There is a related separation worth drawing clearly: infrastructure management is not workload management. A physical data center restricts who can enter the server room while still allowing servers to be administered remotely. Virtual infrastructure should do the same. Restrict vCenter access, and administer the guest operating systems directly over RDP or SSH on their own networks, rather than reaching them through the virtualization management layer and the VM console. That separation pays off in several ways:

- Cleaner security boundaries between infrastructure and workloads, which also means cleaner scopes for compliance audits
- Better visibility, because direct workload access keeps traffic in view of network controls such as VMware vDefend, where console access would bypass them

- A smaller attack surface on the management plane, and time bought to patch when a vulnerability is disclosed
- Simpler firewall rules, because workloads are reached directly rather than through management networks

Restrict Access to Other Connected Infrastructure Systems

Apply the same access discipline to the management interfaces of every connected infrastructure system, not only to VCF.

Counters: lateral movement, backup sabotage and recovery denial, out-of-band management abuse.

VCF does not run alone. Storage arrays, fabric and network switches, backup systems, and hardware management controllers such as iLO, iDRAC, and XClarity all sit in the same blast radius, and each has a management interface that can be abused to reach or damage the environment. An attacker who cannot get into vCenter may be perfectly happy to encrypt the storage array or wipe the backups instead. Restricting access to vCenter while leaving the array's web console reachable from the corporate network moves the problem rather than solving it.

The same principles apply across all of them: limit who can reach each management interface, keep those interfaces off general-purpose networks, and be cautious about enrolling them in centralized directories or connecting them to centralized management tooling that holds write access. Hardware management controllers deserve particular attention, both because they often have side-channel paths to the host, such as virtual USB or NIC emulation, and because they are easy to forget. Backups deserve attention because they are the recovery path, which is exactly why ransomware operators target them first. Keep backup identity isolated from the production directory, so an attacker who takes the corporate directory cannot also reach the backups. Destroying or encrypting the backups before deploying ransomware, to deny recovery and force payment, is now a routine part of these campaigns.

Service Accounts

Give every service account the least privilege it needs, prefer short-lived scoped credentials over standing passwords, and rotate what you cannot replace.

Counters: non-human identity abuse, long-lived secret theft, harvested automation credentials.

A service account is created for an integration, granted broad rights to make it work on the first try, and then left in place for years with a password sitting in a script. It does not log in interactively, so it rarely gets the scrutiny a human account would, and that combination, broad rights plus low visibility plus a long-lived secret, is what makes service accounts a favored target. Attackers hunt for these credentials specifically, scraping them from scripts, configuration files, and code repositories, because one over-permissioned, long-lived account, found once, opens everything it can reach. Long-lived secrets are now recognized as a leading non-human-identity risk in their own right [8].

Apply the same discipline you apply to people, and then some. Scope each service account to exactly the operations it performs and no more; an automation account that powers VMs on and off does not need administrator rights, and the gap between what it has and what it needs is pure risk. Prefer short-lived, narrowly-scoped tokens, or certificate-based authentication with automatic rotation, over long-lived shared secrets, so that a credential issued on demand and expiring quickly is far less useful to an attacker than a password sitting in a deployment script, and one leaked credential is not reused everywhere. Do not press a human user account into service as an automation identity either; use a purpose-built service or workload identity, which is made for the job and keeps a person's access out of a script. Tight scoping matters more here than almost anywhere, because non-interactive identities slip past controls aimed at people: conditional access and MFA policies scoped to users never evaluate a service principal's calls.

VCF 9.1 reduces the need to embed administrator passwords in automation [16]. Through the VCF Identity Broker you register an API client in VCF Operations and issue it a scoped API token; the automation then exchanges that token for a short-lived bearer token, good for around thirty minutes, which it uses across vCenter, NSX, VCF Operations, and VCF Automation. The bearer token is short-lived, but the API token behind it can live much longer, up to thirty days, so it becomes the new high-value secret: store it in a vault, scope it narrowly, rotate it, watch its use, and revoke it when it is no longer needed. The pattern fits most automation where it is supported, but not all of it; some backup tools, older plugins, and third-party

integrations still expect a long-lived credential, so a hybrid approach remains necessary, with those exceptions inventoried, scoped, and rotated.

For the credentials that genuinely cannot yet be replaced with short-lived tokens, default accounts and agent passwords among them, rotate them on a schedule and store them where the rotation can be automated and audited. The goal across all of it is the same: nothing standing, broadly privileged, and long-lived all at once.

Retain and Monitor Identity Logs

Retain identity and access logs for as long as you can fund, and pay attention to successful logins, not only failed ones.

Counters: long attacker dwell time, track covering, delayed detection.

An attacker logging in with valid credentials generates a successful authentication, the one event most environments never alert on. Failed logins get watched closely; the successes, where "they log in, not break in" actually shows up, go unread: a privileged account authenticating at an odd hour, from a new location, or from a device it has never used before.

Retain the access and audit logs from the infrastructure identity provider and from the management plane for as long as your budget allows, because intrusions are often discovered long after initial access and short retention erases the evidence you most need. Google's Mandiant reported an average attacker dwell time of over a year for one recent espionage campaign [9], longer than the log retention many organizations keep, which left some unable to reconstruct how the intrusion began. Keep that long-term history in your log platform or SIEM rather than in the vCenter database, which is not built to hold months of events and will struggle if pushed to. Protect the logs against modification and deletion as well, because altering or purging them to cover tracks is a standard step in an intrusion, not an afterthought.

Feed the logs to monitoring that understands the infrastructure, not only the identity provider. The events worth an alert are the ones that line up with the steps a real intrusion takes: a privileged account authenticating from an unfamiliar address or client, an unexpected host reboot, a change to a group that grants infrastructure access, a new authenticator enrolled on an administrator account. For the management plane, VCF Log Management and VCF Operations can centralize and retain authentication and authorization events; for the identity provider, retain its sign-in and audit logs to the extent the platform allows and forward them to your SIEM. VCF 9.1 adds file integrity monitoring, enabled by default, which checks the static files and binaries that vCenter installs on a regular schedule and streams any unexpected change to VCF Log Management [17], turning quiet tampering with the management plane into a logged event.

The point is not to collect logs for their own sake. It is to make the quiet login loud enough to notice. Retention and monitoring of administrator activity also carry privacy and labor implications, especially across borders, so design them with legal and privacy review (Appendix F).

Rehearse Procedural Controls

Run tabletop exercises that simulate AI-assisted social engineering and credential attacks, and use them to test the human controls, not just the technical ones.

Counters: help-desk social engineering, procedural gaps under pressure.

The technical controls here either work or they do not, and you can verify most of them by inspection: lockdown mode is on or off, federation points where you think it does, a token expires when it should. The procedural controls are different. Whether the help desk actually refuses to reset a privileged authenticator, whether break-glass credentials are reachable by the right people and no one else, whether an administrator under pressure takes the out-of-band path instead of the fast one, none of that is visible until someone tries it. The first real test should not be the actual attack.

Tabletop exercises are how you find the gap before an attacker does. Walk a small group through a concrete scenario: a caller with a cloned voice and a plausible story asks the help desk to reset an infrastructure administrator's authenticator; a "manager" emails after hours demanding emergency access; a monitoring integration is found pushing configuration it should not. Make the scenarios specific to the controls you built, and make them as convincing as the current generation of AI tooling

allows, because that is what the controls will face. The failures these exercises surface tend to be procedural and cheap to fix once seen: a step everyone assumed someone else owned, an approver who is unreachable, a break-glass envelope no one can find. For outsourced or third-party help desks, put the verification standard in the contract and exercise it the same way.

Run them on a schedule, and rerun them after any change to the identity architecture or to the teams that operate it. A control that passed a tabletop a year ago, before a reorganization moved the help desk and three people changed roles, has not been tested recently. It has been assumed.

Isolate Backup and Recovery Systems

Run backup and recovery in their own isolated environment, with their own identity, management, and credentials, not as a corner of the production fleet.

Counters: recovery denial, ransomware targeting backups, shared-management-plane compromise, replication of compromised state, defender lockout.

A recovery environment is worth having only if it survives the compromise of the thing it is meant to recover, and that single test drives the design. Every dependency the recovery environment shares with production, the management plane, the identity domain, the certificate trust, the network path, the update channel, is a path a compromise can travel, and the recovery copy is exactly what an attacker wants to reach, because destroying it is what forces the payment. Ransomware operators now go after identity, the virtualization management plane, and backups together, in one motion, for that reason [1].

A recovery environment that authenticates against the same VCF SSO domain, the same directory, and the same federation as production is reachable by the same credential theft and identity-provider compromise that reaches everything else, and identity is what attackers go after first. Give the recovery side its own identity domain, its own secrets store, and its own break-glass accounts with independent MFA, so that losing production identity does not also lose the console you need to recover at the moment you need it most. Intrusions have emptied whole vaults of privileged credentials in a single session and then forced password changes that locked the defenders out during the crisis [1]; separate recovery identity keeps a working console in your hands when the production one has been taken. Test that this access works under failover conditions and not only on a normal day, because an MFA method that depends on a service or a device identifier that changes during a site switch can lock out the very people running the recovery.

For VCF specifically, VCF Protection and Recovery should target a dedicated VCF deployment with its own management stack, its own vCenter, VCF Operations, and SSO domain, rather than a separate workload domain or cluster inside the same VCF instance. A recovery vCenter joined to the same vCenter Group, or sharing the same VCF Operations scope, is not a separate environment; it is the same environment with a second address, and the intrusion that owns production management owns it too. Give the recovery side its own management stack so the administrative scope ends at the production boundary, with no trusted, routable path from a compromised production console into the place of last resort [14]. Independent administration also means no single operator's access spans both sides, so reaching the recovery environment takes a second compromise or collusion rather than one set of stolen credentials.

Replication deserves the same discipline as administration. A tight recovery point objective faithfully copies whatever it is given, malware mid-encryption, a planted backdoor, or silent corruption included, so replication should flow one direction over a tightly controlled path, recovery points should be immutable and retained long enough to roll back to a known-good state from before the intrusion, and recovery candidates should come up first in an isolated clean room for validation before any failback [14]. These replication and cyber-recovery capabilities of VCF Protection and Recovery, the immutable recovery points and the isolated clean room among them, are licensed through VCF Advanced Cyber Compliance (ACC), a separately licensed advanced service, rather than included in the base platform.

The same separation pays off when no attacker is involved: a faulty update or a fat-fingered bulk operation that rides one shared change window reaches production and recovery at the same instant, so let the recovery side run its own change control and trail production by a deliberate interval.

Regulators are converging on the same architecture, which helps when the audience answers to an auditor. For EU financial entities subject to it, the Digital Operational Resilience Act requires in Article 12 that, when an entity restores backup data using its own systems, those restoration systems be physically and logically segregated from the source [19]. Financial-sector models such as Sheltered Harbor have long called for a data vault that is encrypted, immutable, and fully separated from the institution's own infrastructure [20]. Different bodies and different sectors reach the same conclusion: the recovery environment should be isolated from production rather than centrally managed with it.

Common Objections

The recommendations here, isolation most of all, draw predictable pushback. Every one of these objections is legitimate. They are valid operational concerns, not excuses, and each has a solution for organizations willing to explore it a little. The teams that adopted this architecture did not wave the objections away; they worked through them.

“This Is Not How We Have Done Things”

The concern behind this is real: the current arrangement works, the staff know it, and change carries disruption and risk of its own. It is still worth answering. Attaching infrastructure to the corporate identity provider was a sensible choice for the threats of its time, and those threats have changed. An architecture chosen before identity became the primary target is due for a review, and "we have always done it this way" describes the past rather than making an argument about the present.

“We Have Never Had an Incident”

The instinct is sound, to put scarce effort where there is evidence of risk, but here the evidence misleads. This is survivorship bias, and it is both the most comfortable objection and the most dangerous. A clean record gets read as proof the protections work, when it may instead mean the serious test has not arrived, or that an intruder is present and has not been found. These attacks are quiet by design: they log in with valid credentials rather than breaking anything, they operate at the hypervisor and management layers, where endpoint detection has historically had limited visibility, and the events that would expose them are successful logins that most environments never review. "No incident detected" and "no incident occurred" are different statements, and the gap between them is where a year-long intrusion lives. A clean record is a reason to confirm the defenses are real, not a reason to assume they are.

“Our Administrators Are Not Identity Specialists”

A fair concern, and worth framing correctly. VCF administrators tend to be among the most capable generalists in an IT organization, fluent across compute, storage, networking, and operations, but breadth is not the same as specialization, and running an identity provider well is its own discipline. The answer is to borrow that expertise rather than manufacture it. Some organizations build a small "meta-team" that brings one member of the corporate identity team in to help design and operate the isolated provider alongside the VCF administrators. That person works on the isolated provider through a dedicated infrastructure account on the isolated provider itself, not through any corporate credential, so they remain a corporate-identity administrator on the corporate side and hold a separate infrastructure-identity account on the isolated side, with no single account bridging the two. The identity specialist supplies the operational knowledge while the VCF team keeps ownership of the boundary, and the gap closes without turning the infrastructure staff into identity engineers overnight.

“This Adds Complexity”

Usually it does the opposite. The isolated provider does one job for one small group, so it stays far simpler than the corporate identity system, which carries the accumulated complexity of serving everyone. The population of VCF administrators tends to be small and generally knowledgeable, which keeps both the support load and the chance of misconfiguration low. A small, single-purpose system is easier to secure and audit than a large one serving the whole organization.

“This Creates a Dependency Problem”

Infrastructure has foundational dependencies, DNS, time, identity, and key management among them, and a new identity provider has to fit into that web without creating a circular startup problem. Several worked solutions exist. A single ESX host can act as a dependency host, running DNS resolution, the identity provider, and a key management service as virtual machines that power on first. Because VCF 9.1 federates to any SAML or OIDC provider, that provider need not be elaborate:

some organizations run a small Linux host in each data center with an open-source provider such as Keycloak, BIND for DNS, and ntpd for time, a footprint that would fit on a Raspberry Pi. Microsoft Active Directory covers the same ground with native replication and runs comfortably on a pair of mini desktop machines on a shelf in the rack. You can also run these services as virtual machines in the management domain, with one caveat worth planning for: during an incident you may have to log into their host directly as root to power them on, so pin them to a known host with a DRS should-run rule and you will know where to find them.

"Offboarding and Onboarding Get Harder"

This is true, and the added step is doing real work. Joining or leaving the infrastructure team becomes a deliberate, manual action rather than an automatic consequence of an HR record changing, which is exactly what removes the corporate provisioning pipeline, and HR systems in general, as a path into infrastructure. A provisioning system that can create an infrastructure administrator can also be driven to invent a new hire who does not exist, or to cut off a legitimate administrator at the worst possible moment. The team is smaller, so the extra process should be smaller, and its membership should not be turning over often in any case, since frequent churn on an infrastructure team is its own risk.

"This Costs Money We Have Not Budgeted"

The recurring cost is real, not zero: operating-system licenses, lifecycle operations, monitoring, access reviews, credential recovery, disaster-recovery testing, and audit evidence, plus possibly some hardware, though that need not be extensive. It is still lower than running a general-purpose enterprise identity provider, and the larger spend is the one-time effort to stand the environment up. Set the ongoing cost against what an unbounded identity compromise of the management plane would cost, and the case is straightforward to make.

"The Isolated Provider Has to Be Highly Available"

Infrastructure cannot afford to lose the system it authenticates against. High availability for a small identity provider is well-trodden ground, though. Microsoft Active Directory with AD FS replicates natively and understands sites, and high availability for it is well-documented, though as a Tier 0 dependency it still has to account for certificate lifecycle, service accounts, patching, monitoring, and disaster recovery. A pair of small Linux hosts running keepalived presents a single always-on service to the infrastructure. A single provider in one location, with replicas reachable over WAN links elsewhere, is another common pattern. The requirement is real, and it has more than one good answer.

Bounding the Blast Radius

None of these recommendations stops an attacker who has machine assistance from finding a way in. That is the wrong goal. The goal is that if a corporate identity is compromised, and AI-assisted attacks against systems and people make that more likely than it used to be, the compromise stops at a boundary instead of running to the data center.

Much of the work is subtractive, and the heaviest lifting is one-time: a separate identity provider, fewer connected systems, access narrowed, credentials made short-lived, logs kept and watched, and a recovery environment kept apart. Running what remains is ongoing but bounded, and lighter than operating a general-purpose identity provider. The payoff is a bounded blast radius, which is the difference between an incident and a catastrophe. Patch quickly, segment well, and keep an isolated recovery path tested. This paper covers the identity piece, and it is the piece an attacker reaches for first.

You can tell the architecture is working by a concrete test. It succeeds when a compromise of the corporate identity provider does not grant authentication, authorization, recovery, backup, logging, or credential-reset control over VCF; when privileged access requires phishing-resistant authentication from a managed administrative device; when direct host access is exceptional and logged; when service identities are scoped and rotated; and when the recovery environment stays administratively reachable after production identity is compromised.

Appendix A: Control Priority and Severity

Organizations implement in order of impact, not all at once. The tiers rank controls by how much they bound the blast radius, not by how quickly they can be deployed, which varies by environment; the keywords (must, should, may) indicate strength.

Tier 0, the foundational controls:

- Stand up a separate infrastructure identity provider, with no trust relationship to corporate identity. (should)
- Require phishing-resistant MFA for all privileged access. (must)
- Keep break-glass access independent of the corporate identity provider. (must)
- Remove the corporate help desk's ability to reset infrastructure administrator authenticators. (must)
- Remove direct inbound internet exposure of vCenter, ESX, and other management interfaces. (must)
- Restrict direct host access: ESX lockdown, SSH and local shell off. (must)
- Forward vCenter, ESX, and identity-provider logs off-host to a SIEM. (must)

Tier 1, near-term:

- Separate backup identity from the production directory. (should)
- Full recovery-environment identity separation on a dedicated VCF deployment. (should)
- Privileged-access workstations for administrators. (should)
- Just-in-time privilege and a PAM workflow for the most powerful credentials. (should)
- Inventory and scope service accounts. (should)
- Enable ESX audit logs and forward them. (should)

Tier 2, maturity:

- Tokenized automation with short-lived, scoped credentials. (should)
- Clean-room recovery validation. (should)
- Advanced, hypervisor-aware detections. (should)
- Continuous trust validation and access recertification. (should)
- Non-obvious naming of privileged groups, as a minor reconnaissance-reduction measure only. (may)

Appendix B: Deployment Patterns

Two patterns bracket the range, with a third for provider-operated environments.

Minimum viable, for small or constrained environments:

- Separate administrative accounts for infrastructure, distinct from corporate accounts.
- A dedicated MFA method for infrastructure, phishing-resistant where possible.
- No corporate help-desk reset path for infrastructure accounts.
- A restricted management network, reached through a controlled path.
- ESX lockdown with SSH and local shell off.
- Local break-glass accounts stored securely.
- Log forwarding to a central platform.
- Backup identity separated from the production directory.
- Quarterly access review and an annual tabletop test.

This can run on a lightweight isolated provider, for example an open-source provider with DNS and time on a small dedicated host, rather than a full identity platform.

Advanced, for large or regulated environments, adds to the minimum-viable controls:

- A fully isolated infrastructure identity provider with its own dependencies (Appendix I).
- Privileged-access workstations.
- PAM with just-in-time elevation.
- A dedicated recovery VCF deployment.
- Continuous access evaluation.
- Hypervisor-aware detections (Appendix E).
- The lifecycle and validation programs in Appendices D, F, and K.

Managed service provider or provider-operated:

- Distinct provider-administrator and customer-administrator identities; neither inherits the other's rights by default.
- Delegated, scoped RBAC per customer, with no standing cross-customer administrative reach.
- Tenant separation that bounds one customer's compromise from another's.
- Just-in-time, approved, time-bound, session-recorded support access rather than standing provider accounts.
- A model for both provider-IdP and customer-IdP compromise, where neither grants the other's management plane.
- Contractual MFA, help-desk, reset, and logging obligations on the provider, exercised and audited (Appendix L).

Appendix C: Implementation Path

A workable sequence from a connected starting point to an isolated one:

- Inventory identities, groups, service accounts, local accounts, and trust relationships across the VCF components.
- Map current authentication paths into vCenter, ESX, NSX, VCF Operations, and VCF Automation.
- Build the isolated infrastructure identity provider and its dependencies (Appendix I).
- Create and test break-glass accounts (Appendix G).
- Integrate the VCF Identity Broker against the isolated provider.
- Move a pilot administrator group and validate.
- Enforce phishing-resistant MFA for the pilot, then broadly.
- Migrate service accounts to scoped, short-lived credentials.
- Lock down direct host and appliance access.
- Forward logs and stand up detections (Appendix E).
- Separate backup and recovery identity.
- Test recovery, including recovery while production identity is offline (Appendix H).
- Decommission the old corporate-IdP authentication paths into VCF.
- Run an access review and a tabletop (Appendix S), then set the recurring cadence.

Appendix D: Control Validation, How to Prove It Works

Each control should have a test that demonstrates it, run on a schedule:

- A compromised corporate account cannot authenticate to any VCF component.
- The corporate help desk cannot reset an infrastructure administrator's MFA or password.
- A corporate identity-provider administrator cannot add anyone to a VCF administrative role.
- A new MFA device cannot be enrolled on a privileged account with a password alone.
- Recovery of a lost administrator key requires dual approval.
- Enabling SSH on an ESX host raises an alert.
- A vCenter appliance reboot raises an alert.
- Creating a new API client or long-lived token raises an alert.
- Adding an account to a backup-administrator group raises an alert.
- The recovery environment remains administratively reachable when the production identity provider is offline.
- ESX and vCenter logs are present and intact in the SIEM after a host reboot or a simulated compromise.

Appendix E: Detection and Logging

Logging configuration:

- vCenter events (administrative actions, role and permission changes, logins) are on by default; forward them to a remote log platform or SIEM.
- VCSA appliance operating-system logs (SSH, sudo, local shell, and single-user or bootloader access) are separate from vCenter events and are where a passwordless root shell or an appliance console session shows up; forwarding them off the appliance requires explicit configuration, so enable it.
- ESX standard logs (hostd, vmkernel) carry host operations and are on by default; ESX audit logs are not, so enable the audit logs and forward both off-host.
- Forward identity-provider sign-in and audit logs, and network flow records from the vCenter appliance and ESX hosts.
- Store long-term history in the log platform, not the vCenter database.
- VCF 9.1 adds default-on File Integrity Monitoring for vCenter, every four hours, and supports partner endpoint-detection agents on ESX in isolated containers [17]; use both.

Appliance operating-system changes, including shell-level log forwarding and firewall adjustments, can affect supportability; make them through supported Broadcom mechanisms and the Security Configuration and Hardening Guides [15] rather than ad hoc edits, and confirm the approach for your VCF version.

Detections to build (alert on):

- New MFA device on a privileged account; privileged password reset; any help-desk reset touching an infrastructure identity.
- New federation or trust provider; new enterprise application or service principal; new API client; new long-lived token.
- Privileged group membership change; vCenter role or permission change.
- vCenter appliance console opened; vCenter appliance reboot; unexpected host reboot.
- ESX SSH or shell enabled; lockdown disabled; host added to the exception-user list; ESX audit logging disabled; VIB installed.
- Datastore rename or mass file modification; backup job disabled; backup repository deleted; storage snapshot deletion; retention change.
- Hardware-management-controller login, and remote-console, virtual-USB, virtual-NIC, boot-order, firmware, or password changes on a controller.
- Login from a new ASN, geography, or device; impossible travel; authentication without expected device compliance.
- Unexpected outbound connections from the vCenter appliance or an ESX host, such as an unexplained curl or wget.

Federation drift, review and alert on: new SAML or OIDC applications; changed redirect URIs; new signing certificates; federation-metadata changes; new enterprise applications or service principals; changes to Conditional Access exclusions; application consent and delegated permissions; provisioning connectors; and break-glass exclusions.

Appendix F: Log Retention and Monitoring Governance

Set retention by policy, not by leftover budget. Define minimum hot, warm, and cold retention for identity, vCenter, ESX, NSX, backup, PAM, vault, and recovery logs, driven by expected attacker dwell time (intrusions have run well over a year), incident-reconstruction needs, regulatory obligations, and litigation hold, and bounded by storage cost and privacy constraints. NIST SP 800-92 provides guidance on security log management, and NIST SP 800-63B frames identity-record retention as a policy and risk-management matter [10][21].

Long retention and monitoring of administrator activity carry privacy and labor implications. In multinational environments, logs that hold geolocation, device identifiers, IP addresses, and administrator behavior can engage employee-monitoring law, works-council agreements, and data-residency rules. Design retention and monitoring with legal and privacy review.

Appendix G: Break-Glass Runbook

Define, and test, the following for emergency access:

- How many break-glass accounts exist, and which systems each can reach.
- That they authenticate locally and bypass federation, so they work when the identity provider is down.
- Password length and rotation, and hardware or offline second factors rather than phone-based ones.
- Dual-control retrieval, with the secret stored in an offline safe, sealed envelope, hardware vault, HSM-backed secret, or escrow.
- A test schedule, alerting on any use, and mandatory rotation after every use.
- Documented procedures for when the identity provider is down, when vCenter is down, and when the vault is unavailable.
- Named ownership of break-glass during incident command.

In-person identity proofing is the strongest reset control, but it does not always scale to global, remote, or after-hours teams. Acceptable high-assurance alternatives include two-person video plus hardware-key possession plus manager approval; pre-registered recovery codes in sealed custody; regional trusted custodians; notary-grade proofing; hardware-key escrow; and recovery through PAM with dual approval. In-person should not be the only acceptable path.

Appendix H: Survivable Access, Don't Lock Yourself Out

Secure access is worthless if it is not survivable access. Test administrative reachability under each failure, on a schedule:

- Corporate identity provider offline; infrastructure identity provider offline.
- DNS failure; NTP failure; certificate expiration.
- VCF Identity Broker cluster failure; vCenter down; vCenter appliance root password unknown.
- Lost FIDO2 key; an administrator terminated during an incident.
- Recovery-site failover; SIEM, vault, PAM, or bastion unavailable.
- Network segmentation misapplied.

For each, confirm there is a documented, tested way back in that does not depend on the failed component.

Appendix I: Dependency Isolation

A provider is only as isolated as the services it leans on. Isolation leaks if the infrastructure identity provider depends on corporate DNS, PKI, NTP, certificate automation, key management, email recovery, or endpoint management. Provide:

- Dedicated, resilient DNS for infrastructure identity, and a time source correct enough for Kerberos, SAML, and OIDC to validate.
- A certificate issuance and renewal path separate from the corporate authority, with certificate-expiry monitoring.
- A key-management or HSM availability plan of its own.
- Privileged account recovery kept off corporate email and SMS.
- Separate ownership of the domains, the tenant administrator roles, and the break-glass contact methods.

Appendix J: Cloud Identity-Tenant Isolation Checklist

A separate cloud tenant is not isolated if it still shares the pieces that matter. Confirm it does not share:

- Global or tenant administrators.
- The corporate email or SMS recovery path.
- The same MFA provider or phone numbers.
- The same device-management tenant.
- DNS or domain control.
- The same privileged-access-management platform.
- The same help-desk reset process.
- The same billing or subscription administrators.

A shared global administrator or a shared recovery mailbox undoes the isolation as surely as a shared password would.

Appendix K: Account Lifecycle and Offboarding

A manual joiner-mover-leaver process is the price of isolation, and a missed departure is its main risk. Contain it with:

- Quarterly access reviews at a minimum.
- Expiring privileged memberships and just-in-time elevation rather than standing rights.
- Manager and system-owner attestation.
- Joiner-mover-leaver tickets linked to the actual account changes.
- Last-login review and dormant-account disablement.
- An emergency-termination procedure independent of HR automation.

Appendix L: Help Desk and Credential Recovery

Put the following in policy and, for outsourced desks, in the contract:

- The help desk must not reset infrastructure administrator credentials or authenticators.
- MFA enrollment for privileged infrastructure accounts requires named approvers.
- Voice or video identity proofing alone is insufficient.
- Emergency resets require dual control.
- All reset attempts are logged and sent to the SIEM.
- Failed social-engineering attempts are raised as incident tickets.
- Vendor support cannot bypass the procedure.

Appendix M: Third-Party, Vendor, and Support Access

VCF environments involve platform, OEM, managed-service, backup, storage, network, and hardware vendors. Define:

- A vendor administrative-identity model distinct from staff identities.
- A temporary, approved, time-bound access process with session recording.
- Support-bundle handling and remote-support-tunnel controls.
- Vendor MFA requirements.
- Expiration of vendor accounts.
- An emergency-support path during an outage that is still scoped and logged.
- Audit evidence for all third-party access.

Appendix N: Hardware Management Controllers

Controllers such as iLO, iDRAC, and XClarity need their own treatment:

- A dedicated out-of-band management network.
- No corporate-directory dependency unless that directory is itself isolated.
- Unique local credentials per controller, with backup or escrow of those credentials.
- MFA or bastion-mediated access where supported.
- Firmware lifecycle management and virtual-media restrictions.
- Power-control logging.
- Alerts on remote console, virtual USB, virtual NIC, boot-order changes, firmware changes, and password changes.

Appendix O: Host and Console Access

Direct host and VM-console access are privileged exceptions, not routine paths. The VM console is legitimately needed for a failed network stack or bootloader, initial OS installation, incident containment that isolates the guest, recovery from a firewall mistake, forensic acquisition, and break-glass when workload identity is unavailable. In every case, console access is time-bound, approved, and logged. For host identity, distinguish host authentication, host authorization, host service accounts, vCenter-to-host trust, and local break-glass users; the rule is to avoid the enterprise directory as a direct host-administration path, drive host access through vCenter-mediated RBAC, and strictly control and audit local and break-glass host access.

Appendix P: Privileged and Destructive Actions

Some identity actions are Tier 0: who can change MFA policy, exempt a user from Conditional Access, disable a phishing-resistant requirement, add a new identity provider, modify break-glass accounts, rotate SAML signing certificates, create API clients, or grant consent to applications. Hold these to the same bar as administrator-authenticator resets.

Destructive operations deserve stronger controls than routine ones. Separate the routine roles, read-only inventory, power operations, snapshot operations, datastore operations, host maintenance, and network changes, from the destructive ones: delete VM, delete snapshot, delete backup, change retention, disable a job, modify an identity source, add or remove a host, enable SSH, and change lockdown. Require step-up authentication, approval, or dual control for the destructive set. Draw the same routine-versus-destructive line in each management component that exposes these actions, vCenter, NSX, VCF Operations, VCF Automation, the backup system, the storage array, and the out-of-band controllers, and tie every destructive action to a recorded approval and a log source in the SIEM, so the separation is verifiable rather than nominal.

Appendix Q: Compensating Controls When Isolation Is Not Yet Possible

Some environments cannot separate the identity provider. Until they can, reduce the risk inside the corporate provider:

- Dedicated privileged accounts for infrastructure, in a separate administrative organizational unit or set of groups.
- Phishing-resistant MFA only for those accounts.
- Privileged Identity Management with just-in-time elevation.
- Conditional Access requiring privileged-access workstations.
- No help-desk reset path for those accounts.
- Separate backup identity.
- Authorization assigned in vCenter rather than inherited from broad directory groups.
- Enhanced logging and alerting on those accounts.

Appendix R: Exception Handling

A control that says "never" without an exception process ends up ignored. Define who approves an exception, how long it lasts, what compensating controls it requires, how it is logged, how often it is reviewed, and how emergency exceptions are closed out.

Appendix S: AI-Assisted Tabletop, Legal and Ethical Guardrails

Tabletop scenarios should be realistic without harming staff or breaking the law:

- Do not clone real employees' voices without consent and legal approval.
- Do not collect biometric data unnecessarily.
- Use synthetic personas or approved recordings.
- Coordinate with HR, legal, and privacy.
- Avoid traumatizing help-desk staff.
- Measure process adherence, not individual embarrassment.

Appendix T: Attack-Path Examples

The UNC3944 chain is one of several attack paths worth modeling. Each maps to controls in this guidance:

- Corporate identity-provider compromise, then membership in a vCenter administrative group, then ESX ransomware.
- Help-desk MFA reset, then a VCF administrator login, then backup deletion.
- Service-account token leak, then API automation, then mass VM destruction.
- vCenter vulnerability, then host compromise, then theft of the directory database from a domain-controller VM.
- Backup-console compromise, then retention deletion, then ransomware with no clean recovery.
- Storage-array compromise, then snapshot deletion, then data loss.
- Out-of-band controller compromise, then host boot tampering.
- Shared recovery-site identity, then a production compromise that reaches the recovery environment.

References

1. Mandiant / Google Threat Intelligence Group. M-Trends 2026: Data, Insights, and Strategies From the Frontlines. March 23, 2026. https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2026 (voice phishing ranked the second most common initial infection vector at 11%, and exploits first at 32%, in 2025 intrusion data). Prior-year baseline: M-Trends 2025, April 23, 2025. https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2025
2. Verizon Business. 2026 Data Breach Investigations Report. May 20, 2026. https://www.verizon.com/business/resources/reports/dbir/ (software-vulnerability exploitation began 31% of breaches, overtaking stolen credentials).
3. IBM X-Force. Threat Intelligence Index 2025. April 17, 2025. https://newsroom.ibm.com/2025-04-17-2025-ibm-x-force-threat-index-large-scale-credential-theft-escalates,-threat-actors-pivot-to-stealthier-tactics (abuse of valid accounts tied for the top initial-access vector at 30% of incident-response cases).
4. Google Threat Intelligence Group / Mandiant. "From Help Desk to Hypervisor: Defending Your VMware vSphere Estate from UNC3944." July 23, 2025. https://cloud.google.com/blog/topics/threat-intelligence/defending-vsphere-from-unc3944
5. Cloud Security Alliance (Nathan Montierth). "Scattered Spider: The Group Behind Major ESXi Ransomware Attacks." July 9, 2025. https://cloudsecurityalliance.org/blog/2025/07/09/scattered-spider-the-group-behind-major-esxi-ransomware-attacks
6. FBI, CISA, and partner agencies. "Scattered Spider," Cybersecurity Advisory AA23-320A (updated July 29, 2025). https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a
7. New York State Department of Financial Services. "Industry Letter: Cybersecurity Risks Arising from Artificial Intelligence and Strategies to Combat Related Risks." October 16, 2024. https://www.dfs.ny.gov/industry-guidance/industry-letters/il20241016-cyber-risks-ai-and-strategies-combat-related-risks
8. OWASP Foundation. "OWASP Non-Human Identities Top 10 (2025)." https://owasp.org/www-project-non-human-identities-top-10/2025/top-10-2025/
9. Google Threat Intelligence Group / Mandiant. "Another BRICKSTORM: Stealthy Backdoor Enabling Espionage into Tech and Legal Sectors." September 24, 2025. https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign (393-day average dwell time, attributed to UNC5221).
10. NIST. Special Publication 800-63B-4, Digital Identity Guidelines: Authentication and Authenticator Management (Revision 4). July 2025. https://pages.nist.gov/800-63-4/sp800-63b.html (phishing-resistant authenticators; AAL2 and AAL3; non-exportable keys at AAL3; syncable authenticators do not meet AAL3).
11. CISA. "Implementing Phishing-Resistant MFA" and "Implementing Number Matching in MFA Applications," fact sheets, October 31, 2022. https://www.cisa.gov/sites/default/files/publications/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf and https://www.cisa.gov/sites/default/files/publications/fact-sheet-implement-number-matching-in-mfa-applications-508c.pdf
12. NSA and CISA, Enduring Security Framework. "Identity and Access Management Recommended Best Practices for Administrators" (PP-23-0248). March 21, 2023. https://www.cisa.gov/sites/default/files/2023-12/ESF%20IDENTITY%20AND%20ACCESS%20MANAGEMENT%20RECOMMENDED%20BEST%20PRACTICES%20FOR%20ADMINISTRATORS%20PP-23-0248_508C.pdf
13. Microsoft Entra ID documentation. Seamless SSO: https://learn.microsoft.com/en-us/entra/identity/hybrid/connect/how-to-connect-sso . Self-Service Password Reset policies: https://learn.microsoft.com/en-us/entra/identity/authentication/concept-sspr-policy . Conditional Access: https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview . Continuous Access Evaluation: https://learn.microsoft.com/en-us/entra/identity/conditional-access/concept-continuous-access-evaluation . Emergency-access accounts: https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/security-emergency-access . Managed identities: https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/overview . Workload identities: https://learn.microsoft.com/en-us/entra/workload-id/workload-identities-overview . Privileged Identity Management: https://learn.microsoft.com/en-us/entra/id-governance/privileged-identity-management/pim-configure

15. Broadcom. "Enforcing Strict Identity Separation for Zero Trust." VMware Cloud Foundation Protection and Recovery 9.1 documentation. https://techdocs.broadcom.com/us/en/vmware-cis/vcf/protection-and-recovery/9-1/using-on-premises-ransomware-recovery/guidelines-for-adding-cyber-recovery-to-an-existing-disaster-recovery-site/enforcing-strict-identity-separation-for-zero-trust.html
16. Broadcom. "VMware Cloud Foundation Security Configuration and Hardening Guide." https://github.com/vmware/vcf-security-and-compliance-guidelines
17. Broadcom (Jatin Purohit). "OAuth 2.0 Authentication Using VMware Cloud Foundation PowerCLI 9.1." June 1, 2026. https://blogs.vmware.com/cloud-foundation/2026/06/01/oauth-2-0-authentication-using-vmware-cloud-foundation-powercli-9-1/ (API tokens valid up to roughly 30 days; exchanged bearer tokens expire in roughly 30 minutes).
18. Broadcom (Devyani Pisolkar). "Strengthen Zero Trust Security and Resilience with VCF 9.1." May 5, 2026. https://blogs.vmware.com/cloud-foundation/2026/05/05/platform-security-vcf-9-1/ (default-on File Integrity Monitoring for vCenter, every four hours; partner endpoint detection agents on ESX in isolated containers).
19. Broadcom (Bob Plankers). "Security in VMware Cloud Foundation 9.0." August 5, 2025. https://blogs.vmware.com/cloud-foundation/2025/08/05/security-vmware-cloud-foundation-9-0/
20. European Parliament and Council. Regulation (EU) 2022/2554 (Digital Operational Resilience Act), Article 12(3). December 14, 2022. https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng
21. Sheltered Harbor (FS-ISAC). Financial-sector data-vaulting and resilience model (planned wind-down; assets transferring to FS-ISAC). https://www.fsisac.com/who-we-are/sheltered-harbor
22. NIST. Special Publication 800-92, Guide to Computer Security Log Management. September 2006. https://csrc.nist.gov/pubs/sp/800/92/final