

Identity Security in the Age of AI

Arvind Mishra

Apateon Press

Copyright © 2026 by Arvind Mishra

All rights reserved.

No portion of this book may be reproduced in any form without written permission from the publisher or author, except as permitted by U.S. copyright law.

This book was created to demonstrate Manuscript Partners' AI-assisted writing techniques. The author, Arvind Mishra, is a fictional person, and his biography, experiences, and anecdotes are invented. Any resemblance to a real person, living or dead, is purely coincidental. The organizations, standards, incidents, and publications cited in the footnotes are real and were accurate as of October 2026.

Contents

Introduction: Identity Becomes the New Security Perimeter	1
1. The Evolution of Identity Security	10
2. The Identity-Based Attack Surface	25
3. Identity Security Architecture for the Modern Enterprise	43
4. Authentication and the Future of Digital Trust	60
5. Authorization, Least Privilege, and Access Governance	78
6. Privileged Access and Secrets Management	95
7. Identity Threat Detection, Response, and Recovery	112
8. Non-Human Identities: The Enterprise Security Challenge	129
9. Securing AI Agents and Autonomous Systems	148
10. AI-Powered Attacks and Emerging Identity Threats	166
11. Using AI to Strengthen Identity Security	184
12. Building and Operating an Effective Identity Security Program	201

Introduction: Identity Becomes the New Security Perimeter

Early in my career, I sat in a conference room with the security leadership of a large enterprise as they walked through the aftermath of an intrusion. The attackers had not broken through a firewall or exploited an obscure vulnerability in a server. They had logged in. A contractor's credentials, harvested through a convincing phishing message, gave them a foothold, and from there they moved patiently through the environment using accounts and permissions that, on paper, were entirely legitimate. Every system had done exactly what it was designed to do. The problem was that nobody had a clear answer to a deceptively simple question: who, or what, was actually acting on the network, and should it have been allowed to do what it did?

I have returned to that question many times over the past two decades. I have encountered versions of it while working with global consultancies and while serving in cybersecurity roles inside some

of the largest companies in the world. The details change from one organization to the next, but the underlying pattern has remained remarkably consistent. In my experience, a significant share of serious security incidents eventually trace back to identity: a stolen credential, an over-privileged account, a forgotten service account, a session token that outlived its purpose, or an approval process that existed in policy but not in practice.

This book is my attempt to share what I have learned about that problem, and to explain why I believe it is about to become considerably harder.

Why Identity Now Sits at the Center

For much of the history of enterprise computing, security was organized around a perimeter. We built walls around the data center, controlled what came in and out, and generally assumed that anything inside the boundary could be trusted to some degree. That model was never perfect, but it reflected how organizations actually operated. Employees worked in offices, applications ran on servers the company owned, and the network had a reasonably clear edge.

That edge has largely dissolved. Cloud platforms, SaaS applications, remote and hybrid work, mobile devices, partner integrations, and APIs have distributed enterprise computing across environments that no single organization fully controls. In most of the enterprises I have worked with, there is no longer a meaningful "inside" to protect. What remains, and what increasingly determines whether an action is permitted, is identity. The question has shifted from "Where is this request coming from?" to "Who or what is making this request, how confident are we in that claim, and what should it be allowed to do?"

Attackers understood this shift early, and in many cases faster than defenders did. Why spend effort developing an exploit when a valid username, password, or token will open the door? Credential theft, session hijacking, MFA fatigue attacks, and abuse of excessive permissions have become routine techniques because they work, and because activity performed with legitimate credentials tends to blend into normal operations. In my view, identity has become both the most important control point in the modern enterprise and one of its most frequently targeted weaknesses.

How AI Changes the Equation

If identity were simply growing in importance, the challenge would be significant but familiar. What makes this moment different is artificial intelligence.

AI is changing the identity landscape in at least three ways, and each of them receives substantial attention in this book.

First, AI is changing who, or what, holds access. Organizations are deploying AI agents that can read email, query databases, call APIs, write and execute code, open tickets, and take action across enterprise systems on behalf of users. These agents need identities and permissions in order to function. They often operate continuously, at machine speed, and with a degree of autonomy that traditional access models were never designed to accommodate. When an agent acts, we need to know whose authority it is exercising, what it is permitted to do, and how we would detect and stop it if it began doing something it should not.

Second, AI is strengthening attackers. Phishing messages that once contained telltale errors can now be fluent, personalized, and produced at scale. Voice and video deepfakes are making it more difficult

to trust what we hear and see during help desk calls, executive approvals, and identity verification. Synthetic identities and automated credential attacks are lowering the cost of attempting fraud and intrusion. Many of the human signals we have historically relied on to establish trust appear to be weakening.

Third, AI offers defenders genuinely useful new capabilities. Machine learning can help surface anomalous access patterns, identify dormant and excessive privileges, correlate identity signals across systems that rarely talk to one another, and accelerate investigations that once took days. I am optimistic about these capabilities, but I have also seen enough technology waves to be cautious. AI-assisted security introduces its own risks, including unreliable outputs, data exposure, adversarial manipulation, and the temptation to automate decisions that still warrant human judgment. Using AI well in identity security requires the same discipline we would apply to any other powerful tool.

Taken together, these developments suggest that the relationship between identity, access, and trust is being renegotiated. Identity is no longer only a mechanism for letting people log in. It is becoming the primary means by which an organization decides what can act within its environment, what it can reach, and what it is permitted to do.

From Human-Centric IAM to Human and Machine Identities

Most identity and access management programs were built around people. We onboard employees, assign them roles, give them access to applications, review that access periodically, and remove it when they leave. Even that process, which sounds straightforward, is difficult to execute well in a large organization. I have seen enterprises with tens

of thousands of applications, multiple directories inherited through acquisitions, and access review campaigns that consumed months of effort while producing limited improvement in actual risk.

Yet in many environments, human identities are now outnumbered by non-human identities, sometimes by a wide margin. Service accounts, workloads, containers, bots, API keys, certificates, and automation scripts all authenticate and act within enterprise systems. Many of them hold broad and long-lived privileges. Many have no clearly assigned owner. Some were created years ago for a project that no longer exists, and nobody is entirely sure what would break if they were disabled. On more than one occasion, I have watched a discovery exercise uncover far more machine credentials than the security team expected, along with secrets embedded in code repositories and configuration files that had been accessible far longer than anyone realized.

AI agents add a new category on top of this already complicated picture. They behave in some respects like machine identities and in other respects like delegated human users. They may be able to reason about their tasks, choose which tools to call, and interact with other agents. Our existing frameworks for governance, authentication, and accountability were not designed with these actors in mind, and I believe that adapting those frameworks will be one of the defining security challenges of the coming years.

What I Have Learned in the Enterprise

There is no shortage of material describing the principles of good identity security. Most experienced practitioners already understand least privilege, Zero Trust, multifactor authentication, and access governance. In my experience, the harder questions are rarely about principles. They are about implementation.

How do you enforce least privilege across thousands of applications when the business owners of those applications cannot tell you what access their users actually need? How do you roll out phishing-resistant authentication when a meaningful portion of your workforce uses shared devices, works in plants or call centers, or depends on legacy systems that do not support modern protocols? How do you rotate the credentials of a service account that supports a revenue-critical process when nobody is confident about its dependencies? How do you persuade an organization to invest in identity governance when the benefits are largely invisible until something goes wrong?

These are the questions that have occupied much of my working life, and they shape the perspective of this book. I have seen well-designed security initiatives stall because they underestimated organizational complexity. I have seen expensive tools deliver limited value because they were deployed without the processes, data quality, or ownership needed to make them effective. I have also seen relatively modest, well-prioritized changes produce meaningful and measurable reductions in risk. The difference, more often than not, came down to understanding the environment honestly and making deliberate trade-offs.

My aim is therefore not to present an idealized architecture that assumes unlimited budgets and a blank slate. Few of us have either. Instead, I want to share what tends to work in real enterprises, where it tends to break down, and how security leaders and practitioners can make better decisions under realistic constraints.

How This Book Is Organized

The book is divided into four parts that move from understanding the problem to building and operating a program that addresses it.

Part I examines the new identity security landscape. It traces the evolution from perimeter-based defense to identity-centric security, maps the ways attackers exploit identity across the attack chain, and describes the architectural components that underpin identity security in a modern hybrid enterprise.

Part II focuses on the core foundation: authentication, authorization and access governance, privileged access and secrets management, and identity threat detection and response. These capabilities are not new, but I believe that many organizations still have significant gaps in them, and the arrival of AI makes those gaps more consequential.

Part III turns to the challenges that are specific to the age of AI. It addresses non-human identities, the security of AI agents and autonomous systems, the emerging generation of AI-powered attacks, and the opportunities and risks of applying AI to identity security itself.

Part IV brings these threads together into practical guidance for building and operating an effective identity security program, including assessing maturity, prioritizing investments, developing a roadmap, measuring results, and avoiding the implementation failures I have seen most often.

Throughout the book, you will find several recurring features. Enterprise Field Notes offer short observations drawn from my own experience. Best Practices That Work summarize implementation guidance that has proven effective. Architecture Decisions and Trade-Offs compare alternative approaches and their consequences. Practitioner Checklists provide questions and priorities you can apply directly to your environment. Realistic Threat Scenarios illustrate how identity weaknesses tend to be exploited and how layered controls can detect or contain an attack.

How to Use This Book

I have written this book for CISOs, IT leaders, security architects, identity and access management professionals, and the hands-on practitioners who make identity security work every day. Readers will likely approach it from different directions.

If you are responsible for strategy and investment, you may find it useful to begin with Part I and Part IV, which frame the risk landscape and the program-level decisions that follow from it. If you are an architect or engineer, Parts II and III provide the technical depth on specific capabilities and on the new problems posed by machine identities and AI agents. If you are early in building an identity security program, reading the book from beginning to end should give you a coherent picture of how the pieces fit together.

Whatever your role, I would encourage you to read with your own environment in mind. Use the checklists and assessment questions to identify where your organization stands today, where the most significant gaps lie, and which improvements are likely to deliver the greatest reduction in risk for the effort involved. No organization can address everything at once, and the most successful programs I have seen were the ones that chose their priorities carefully and pursued them consistently.

A Foundation Worth Getting Right

We are entering a period in which a growing share of the work inside our organizations will be performed by systems that act on our behalf. That prospect is exciting, and I expect it to deliver real value. It also means that the questions I first encountered in that conference room years ago will be asked more often, at greater scale, and with higher

stakes. Who or what is acting? On whose authority? With what permissions? And how would we know if something had gone wrong?

I believe identity is where those questions will be answered. Getting it right will not make an enterprise invulnerable, but it may well be the most important step an organization can take toward operating securely in an increasingly autonomous world. I hope this book helps you take that step with greater clarity and confidence.

1. The Evolution of Identity Security

When I began working in enterprise IT, identity was largely an administrative matter. Someone in human resources sent a request, someone in IT created an account, and the new employee received a username, a password, and access to whatever shared drives and applications their manager asked for. Security, in the sense that most executives understood it, lived elsewhere: in the firewalls, the network segmentation, the antivirus software, and the physical security of the data center. Identity was plumbing. It had to work, but few people considered it strategic.

That view has changed dramatically, and in my opinion it has not changed fast enough in many organizations. To understand why identity now sits at the center of enterprise security, and why the arrival of AI agents is likely to push it further toward the center, it helps to look at how we arrived here. This chapter traces that evolution: from perimeter-based security to identity-centric defense, through the maturing of identity and access management as a discipline, into the cloud and SaaS era, and finally to the rapid growth of non-human identities and autonomous agents. Along the way, I will try to explain why the models many organizations still rely on were reasonable for

their time but are increasingly mismatched to the environments we now operate.

From the Castle to the Network Without Walls

For roughly three decades, the dominant model of enterprise security resembled a castle. The organization's valuable assets sat inside a protected boundary, and the security team's primary job was to defend that boundary. Firewalls controlled traffic at the edge. Demilitarized zones hosted the few systems that needed to face the internet. Virtual private networks extended the boundary to remote users by bringing them, logically, inside the walls.

This model rested on an implicit assumption: location was a reasonable proxy for trust. A request originating from inside the corporate network was presumed to come from an employee or an approved system, and it was granted a meaningful degree of trust as a result. Internal networks were often flat, internal applications frequently relied on network location rather than strong authentication, and once a user or device was inside, it could typically reach a great deal.

It is easy to criticize this approach in hindsight, but it made sense given the conditions of the time. Most applications ran in data centers the organization owned. Most employees worked at desks connected to the corporate network. The number of external connections was limited and relatively easy to enumerate. Under those conditions, defending the boundary was a sensible way to concentrate scarce security resources.

The weaknesses of the model became apparent well before the cloud era. Once an attacker gained a foothold inside the perimeter, through a compromised workstation, a malicious insider, or a stolen VPN credential, the implicit trust granted to internal traffic worked

in the attacker's favor. Many of the significant breaches of the 2000s and 2010s followed a similar pattern: initial access through a relatively modest weakness, followed by extended lateral movement through an internal environment that offered little resistance.

Security practitioners recognized this problem. In the mid-2000s, the Jericho Forum argued that organizations should prepare for "de-perimeterization," a world in which the network boundary would no longer provide meaningful protection.¹ Around 2010, Forrester analyst John Kindervag articulated the Zero Trust model, which proposed that no network location should be inherently trusted.² A few years later, Google described its BeyondCorp initiative, which moved access decisions away from the network and toward the identity of the user and the state of the device.³ In 2020, NIST published Special

-
1. The Open Group Jericho Forum, *Jericho Forum Commandments*, Version 1.2, The Open Group, San Francisco, May 2007.
 2. John Kindervag, *No More Chewy Centers: Introducing the Zero Trust Model of Information Security*, Forrester Research, Inc., Cambridge, Mass., September 14, 2010 (updated September 17, 2010), <<https://www.forrester.com/report/No-More-Chewy-Centers/RES56682>>, accessed October 22, 2026.
 3. Rory Ward and Betsy Beyer, "BeyondCorp: A New Approach to Enterprise Security," *login:*, Vol. 39, No. 6, December 2014, pp. 6-11, <<https://www.usenix.org/publications/login/dec14/ward>>, accessed October 22, 2026.

Publication 800-207, which gave Zero Trust architecture a widely referenced formal definition.⁴

The common thread across these efforts was a shift in the question being asked. The perimeter model asked, "Is this request coming from inside or outside?" The emerging model asks, "Who or what is making this request, how confident are we in that claim, is the context appropriate, and what specifically should this entity be allowed to do?" That is fundamentally an identity question.

ENTERPRISE FIELD NOTES

In one large organization I worked with, a security assessment found that a significant number of internal applications performed no authentication of their own. They had been built on the assumption that anyone able to reach them on the internal network was an authorized employee. That assumption may have been defensible when the applications were written. By the time of the assessment, the internal network was reachable by contractors, partner connections, and thousands of remote users over VPN. Nobody had made a decision to extend trust to all of those parties. It had simply accumulated as the environment changed around applications that did not.

4. Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connolly, *Zero Trust Architecture*, NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, Md., August 2020, <<https://doi.org/10.6028/NIST.SP.800-207>>, accessed October 22, 2026.

The Maturing of Identity and Access Management

While the perimeter model dominated security thinking, identity and access management was evolving in parallel, largely as an operational discipline focused on efficiency and administration.

Early enterprise systems each maintained their own user stores. A typical employee might have separate credentials for the mainframe, the email system, the ERP platform, and a dozen departmental applications. This arrangement was inefficient for users and administrators, and it created obvious security problems. When an employee left the company, removing their access meant finding and disabling every one of those accounts, and in practice some were usually missed.

Directory services were the first major step toward consolidation. Standards such as X.500 and the Lightweight Directory Access Protocol (LDAP) made it possible to maintain centralized repositories of user information. Kerberos, developed at MIT, provided a robust protocol for network authentication.⁵ Microsoft's Active Directory, introduced with Windows 2000, combined directory services, Kerberos authentication, and group-based policy management in a form that became deeply embedded in most enterprises. In many of the organizations I have worked with, Active Directory remains a central pillar of the identity infrastructure today, and it also remains one of the most attractive targets for attackers who gain internal access.

The next phase brought formal identity management practices. Provisioning systems automated the creation and removal of accounts

5. Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," *Proceedings of the Winter 1988 USENIX Conference*, USENIX Association, Dallas, February 1988, pp. 191-202.

based on authoritative sources such as human resources systems. Role-based access control offered a way to grant access according to job function rather than individual request. Web access management and early single sign-on products reduced the number of credentials users needed to manage. Regulatory pressure, particularly following the Sarbanes-Oxley Act in the United States, pushed organizations toward periodic access certification, in which managers attest that their employees' access remains appropriate.⁶

Federation standards extended identity beyond organizational boundaries. SAML 2.0, ratified in 2005, allowed one organization to vouch for a user's identity to another.⁷ OAuth 2.0 introduced a widely adopted framework for delegated authorization, enabling an application to act on a user's behalf with limited permissions.⁸ OpenID

6. *Sarbanes-Oxley Act of 2002*, Pub. L. No. 107-204, 116 Stat. 745, July 30, 2002.

7. Scott Cantor, John Kemp, Rob Philpott, and Eve Maler, eds., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, OASIS, March 15, 2005, <<https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>>, accessed October 22, 2026.

8. Dick Hardt, ed., *The OAuth 2.0 Authorization Framework*, RFC 6749, Internet Engineering Task Force, October 2012, <<https://www.rfc-editor.org/rfc/rfc6749>>, accessed October 22, 2026.

Connect added an identity layer on top of OAuth.⁹ These protocols are now foundational to how users authenticate to cloud and SaaS services, and, as later chapters will discuss, they are increasingly relevant to how AI agents are granted access as well.

It is worth noting what this generation of IAM was primarily designed to accomplish. Its goals were largely administrative efficiency, user convenience, and regulatory compliance. Security was certainly a consideration, but IAM teams were frequently organized under IT operations rather than security, and their success was often measured by how quickly they could onboard new employees or complete certification campaigns. In my experience, that heritage still shapes how many organizations fund, staff, and measure their identity programs, and it can create a gap between how identity is managed and how it is actually attacked.

9. Nat Sakimura, John Bradley, Michael B. Jones, Breno de Medeiros, and Chuck Mortimore, *OpenID Connect Core 1.0*, OpenID Foundation, February 25, 2014, <https://openid.net/specs/openid-connect-core-1_0-final.html>, accessed October 22, 2026.

Cloud, SaaS, and the Expanding Identity Landscape

The migration to cloud infrastructure and SaaS applications accelerated the decline of the perimeter and transformed the scope of identity management.

When an organization moves its email, collaboration, customer relationship management, and human resources functions to SaaS providers, those applications no longer sit behind its firewall. Users reach them directly over the internet, from corporate offices, homes, airports, and coffee shops, using managed and unmanaged devices alike. The only consistent control point between the user and the application is the authentication and authorization process. Identity becomes, quite literally, the perimeter.

Infrastructure as a service introduced a parallel shift for technical teams. In a public cloud environment, the ability to create servers, modify network rules, access storage, and change security configurations is governed by the cloud provider's identity and access management system. A misconfigured cloud IAM policy can expose an organization's data as effectively as an open firewall port once did, and in some cases more broadly. Cloud IAM models also tend to be considerably more granular and complex than traditional enterprise directories, with thousands of possible permissions across hundreds of services.

Most large enterprises do not live entirely in the cloud. They operate hybrid environments in which on-premises directories, legacy applications, multiple cloud platforms, and hundreds or thousands of SaaS applications coexist. Each of these environments may have its own identity model, its own administrative roles, and its own logging conventions. Mergers and acquisitions add further layers. I have

worked with organizations that maintained several separate directory forests, multiple identity providers, and identity data spread across systems that had never been designed to work together.

Remote and hybrid work, which expanded sharply during the COVID-19 pandemic, reinforced these trends. Many organizations that had planned gradual transitions to identity-centric access found themselves compressing years of change into months. Some of the shortcuts taken during that period, such as broad exceptions to access policies or hastily configured remote access, may still be present in enterprise environments today.

BEST PRACTICES THAT WORK

Treat your identity provider and directory infrastructure as tier-zero assets, on par with domain controllers and core network infrastructure. In practice, this tends to mean restricting administrative access to a small number of dedicated, hardened accounts; requiring phishing-resistant authentication for those accounts; monitoring configuration changes closely; and including identity infrastructure explicitly in incident response and recovery planning. Many organizations protect the systems that identity grants access to more carefully than they protect identity itself.

The Rise of Non-Human Identities

Until relatively recently, when people discussed identity management, they generally meant the management of human users. That framing has become increasingly incomplete.

Enterprise environments have always contained non-human identities. Service accounts allowed applications to connect to databases. Batch jobs ran under dedicated credentials. Network devices authenticated to management systems. However, the number of these identities remained manageable, and they tended to change slowly.

Several developments have caused non-human identities to multiply. Microservice architectures decompose applications into many small components, each of which may need its own credentials to communicate with others. Containers and serverless functions are created and destroyed continuously, and each instance may require an identity. DevOps pipelines need credentials to build, test, and deploy code across environments. SaaS integrations rely on API keys and OAuth tokens to exchange data. Robotic process automation introduced software bots that log into applications much as a human user would. Infrastructure as code means that the creation of cloud resources, including identities, is itself automated.

Industry estimates of the ratio between non-human and human identities vary widely, and I would treat any specific figure with some caution. What I can say from direct experience is that in every large enterprise where I have seen a serious inventory effort, non-human identities substantially outnumbered human ones, and the security team's prior estimate was too low.

Non-human identities present a distinct set of challenges. They often hold broad privileges because it was easier to grant wide access than to determine precisely what was needed. Their credentials are frequently long-lived, because rotating them risks breaking production systems. They may be embedded in source code, configuration files, scripts, or CI/CD variables. They often lack a clearly accountable owner, particularly when the engineer or team that created them has moved on. And they do not fit neatly into governance processes

designed around people: a service account has no manager to approve its access, does not leave the company, and will not respond to a certification email.

Attackers have taken notice. Compromised API keys, leaked cloud credentials, and abused OAuth tokens have played a role in a growing number of publicly reported incidents. Because non-human identities often operate without interactive logins, multifactor authentication, or behavioral baselines comparable to human users, their misuse can be difficult to detect.

Autonomous Agents: A New Kind of Actor

AI agents represent the most recent, and potentially the most consequential, addition to the identity landscape.

An AI agent, in the sense I use throughout this book, is a system that uses one or more AI models to pursue a goal by planning steps, invoking tools, and taking actions across other systems, often with limited human intervention for each step. An agent might triage a support queue, reconcile financial records, investigate a security alert, write and deploy code, or schedule meetings by reading and responding to email.

To do any of this, an agent needs access. It must authenticate to the systems it interacts with, and it must be authorized to perform specific actions. In that sense, an agent is a non-human identity. But agents differ from traditional service accounts in several important ways. Their behavior is not fully determined by code written in advance; it emerges from a model's interpretation of instructions and context, which means it may vary from one execution to the next. They frequently act on behalf of a particular human user, inheriting or borrowing that user's authority, which blurs the line between human

and machine identity. They may choose which tools to invoke and which data to retrieve. They may delegate tasks to other agents. And they can be influenced by the content they process, including content an attacker has crafted to manipulate them.

These characteristics raise questions that existing identity models were not designed to answer. When an agent takes an action, whose identity should appear in the audit log: the agent's, the user who invoked it, or both? How should permissions be scoped for a system whose specific actions cannot be fully predicted? How do we revoke access for an agent that is running continuously across multiple systems? Who is accountable when an agent acting within its granted permissions does something harmful? Chapters 8 and 9 address these questions in depth. For now, the important point is that agents are arriving in enterprise environments quickly, and in many organizations they appear to be arriving faster than the governance needed to manage them.

ENTERPRISE FIELD NOTES

In conversations with security leaders over the past year, I have heard a recurring theme: AI agents and AI-powered integrations are often introduced by business units, developers, or individual employees before the security team is aware of them. An integration that connects an AI assistant to a corporate email account or document repository may be authorized through a simple OAuth consent screen. From the user's perspective, they clicked "Allow." From the security team's perspective, a new identity with access to sensitive data has appeared without review. Discovering and inventorying these connections is frequently the first practical challenge organizations face.

Why Traditional Identity Security Models Are No Longer Sufficient

Taking these developments together, I believe the traditional approach to identity security falls short in five important respects.

First, traditional models assume that authentication is a single event. A user logs in once in the morning and is trusted for the rest of the day. In an environment where session tokens can be stolen and replayed, where devices change state, and where agents act continuously, trust needs to be evaluated more often and in light of changing context.

Second, traditional models are human-centric. Governance processes, lifecycle management, access reviews, and authentication controls were designed for people with managers, job titles, and employment dates. They map poorly onto service accounts, workloads, and agents, which now represent a large and growing share of the identities in most environments.

Third, traditional models separate identity administration from security operations. IAM teams provision access; security operations teams monitor for threats. Attackers, however, move fluidly between the two domains, abusing legitimate access in ways that may look like normal administrative activity. Effective defense increasingly requires that identity data inform detection and response, and that detection findings feed back into access decisions.

Fourth, traditional models tend to grant standing privileges. Access is assigned and then persists until someone removes it, which in practice often means indefinitely. Standing privilege expands the impact of every compromised identity. Approaches such as just-in-time access

and ephemeral credentials, discussed in Chapters 5 and 6, aim to reduce that exposure.

Fifth, traditional models assume a relatively stable and knowable environment. They depend on accurate inventories of users, applications, and entitlements. In hybrid, multi-cloud, SaaS-heavy environments with rapidly multiplying machine identities and agents, achieving that visibility is itself a substantial undertaking, and many organizations have not yet accomplished it.

None of this means that existing IAM investments are wasted. Directories, identity providers, provisioning systems, and governance platforms remain essential. The challenge is to extend and adapt them so that identity functions not only as an administrative capability but as a core security control, one that spans human and non-human identities and operates continuously rather than periodically.

PRACTITIONER CHECKLIST

Consider the following questions as a starting point for assessing where your organization stands:

Do you have a reasonably complete inventory of human identities across all directories, identity providers, and major applications?

Do you have any meaningful inventory of non-human identities, including service accounts, API keys, OAuth applications, and workload identities? Who owns each one?

Is your identity infrastructure itself treated and protected as a tier-zero asset?

Do identity signals, such as authentication events, privilege changes, and new OAuth grants, flow into your security operations and detection capabilities?

How much of the privileged access in your environment is standing rather than granted on demand?

Do you know which AI agents or AI-powered integrations currently have access to corporate systems and data, and under whose authority they operate?

Looking Ahead

The evolution described in this chapter has moved identity from the margins of security to its center. The perimeter has not entirely disappeared; network controls remain valuable, and I would not recommend abandoning them. But the decisive questions about trust are increasingly answered by identity systems rather than network boundaries.

That shift has made identity an extremely attractive target. In the next chapter, we examine the identity-based attack surface in detail: how attackers obtain and abuse credentials, tokens, and sessions; how identity serves as both an initial access vector and a pathway to privilege escalation; how AI is enhancing social engineering and impersonation; and how these techniques combine into attack chains that can turn a single compromised identity into an enterprise-wide incident. Understanding how identity is attacked is, in my experience, the most reliable foundation for deciding how to defend it.

2. The Identity-Based Attack Surface

Several years ago, I was asked to help review an incident at an organization that had invested heavily in security. It had a modern firewall estate, endpoint detection on every managed device, a well-staffed security operations center, and a respectable vulnerability management program. None of those controls had failed, exactly.

The attacker had obtained the credentials of an IT support employee, used them to sign in to a remote access portal, and then spent the following weeks quietly exploring the environment, reading documentation, enumerating group memberships, and eventually finding an administrative account whose password had not changed in years. By the time the activity was detected, the attacker had created accounts of their own and granted them broad privileges. Almost every action they took had been performed with valid credentials, through approved tools, in ways that resembled ordinary administrative work.

That incident was not unusual. If anything, it was typical. Over the course of my career, I have come to believe that most defenders think

about attacks in terms of vulnerabilities and malware, while many attackers increasingly think in terms of identities and permissions. This chapter attempts to close that gap. It examines how attackers obtain and abuse credentials, tokens, and sessions; how identity serves as both an initial access vector and a pathway to privilege escalation; how AI is sharpening social engineering and impersonation; how these techniques combine into attack chains; and how security leaders can map identity threats to the business risks that executives and boards care about.

Why Attackers Prefer Identity

From an attacker's perspective, a valid identity offers several advantages over a technical exploit.

First, it is often cheaper to obtain. Developing or purchasing a reliable exploit for a well-maintained system can be costly, and patches eventually close the opportunity. Credentials, by contrast, are available through phishing, through malware that harvests stored passwords and browser cookies, through data breaches at other organizations where users reused passwords, and through criminal marketplaces in which initial access to corporate environments is bought and sold.

Second, activity performed with legitimate credentials is difficult to distinguish from legitimate use. Security tools are generally good at identifying known malicious files and suspicious exploit behavior. They are less effective at recognizing that a real administrator account, signing in through a real portal and running real administrative tools, is being operated by the wrong person.

Third, identity tends to provide a direct path to what the attacker wants. Data, financial systems, cloud infrastructure, and source code

are all protected primarily by access controls. Whoever holds the right identity with the right permissions can usually reach them without needing to defeat any additional technical barrier.

Industry incident reports have consistently identified stolen or misused credentials as one of the most common factors in breaches,¹ and the MITRE ATT&CK framework treats the use of valid accounts as a technique relevant to multiple stages of an intrusion, including initial access, persistence, privilege escalation, and defense evasion.² Specific percentages vary from report to report and year to year, so I would not place too much weight on any single figure. The overall pattern, however, has been remarkably stable.

Credentials, Tokens, and Sessions

When we talk about attacks on identity, it helps to be precise about what is being attacked. An identity is represented in practice by several different artifacts, each of which can be stolen or abused.

Passwords and Shared Secrets

Passwords remain the most familiar target. Attackers obtain them through phishing pages that imitate legitimate sign-in screens,

-
1. Verizon Business, *2025 Data Breach Investigations Report*, Verizon, April 2025, <<https://www.verizon.com/business/resources/reports/dbir/>>, accessed October 22, 2026.
 2. The MITRE Corporation, "Valid Accounts, Technique T1078," *MITRE ATT&CK*, <<https://attack.mitre.org/techniques/T1078/>>, accessed October 22, 2026.

through information-stealing malware, and through breaches of third-party services. Credential stuffing attacks take username and password pairs leaked from one site and test them automatically against others, exploiting the persistent habit of password reuse. Password spraying takes the opposite approach, trying a small number of common passwords against a large number of accounts in order to avoid lockout thresholds. Both techniques are inexpensive, highly automated, and, in my experience, still successful far more often than they should be, particularly against accounts that are exempt from multifactor authentication.

Those exemptions deserve particular attention. In almost every large environment I have assessed, some accounts were excluded from MFA requirements, usually for reasons that seemed sensible at the time: a legacy application that could not support modern authentication, a service account used by an integration, an executive who found the prompts inconvenient, or a test environment that was never expected to matter. Attackers actively search for these gaps. Several high-profile incidents in recent years have been publicly attributed,

at least in part, to accounts that lacked MFA, including legacy VPN accounts and cloud accounts used in test environments.³

Multifactor Authentication and Its Limits

Multifactor authentication dramatically raises the cost of credential-based attacks, and I would never discourage an organization from deploying it. However, not all MFA is equally resistant to attack, and attackers have adapted.

MFA fatigue, sometimes called push bombing, involves repeatedly triggering push notifications to a user's phone until the user approves

-
3. Joseph Blount, Testimony before the U.S. Senate Committee on Homeland Security and Governmental Affairs, "Threats to Critical Infrastructure: Examining the Colonial Pipeline Cyber Attack," Washington, D.C., June 8, 2021, <<https://www.hsgac.senate.gov/imo/media/doc/Testimony-Blount-2021-06-08.pdf>>, accessed October 22, 2026; Microsoft Threat Intelligence, "Midnight Blizzard: Guidance for Responders on Nation-State Attack," *Microsoft Security Blog*, January 25, 2024, <<https://www.microsoft.com/en-us/security/blog/2024/01/25/midnight-blizzard-guidance-for-responders-on-nation-state-attack/>>, accessed October 22, 2026; Mandiant, "UNC5537 Targets Snowflake Customer Instances for Data Theft and Extortion," *Google Cloud Blog*, June 10, 2024, <<https://cloud.google.com/blog/topics/threat-intelligence/unc-5537-snowflake-data-theft-extortion>>, accessed October 22, 2026.

one, either by mistake or simply to make the notifications stop.⁴ The attacker may combine this with a message or phone call posing as IT support. SMS-based codes are vulnerable to SIM swapping, in which an attacker persuades a mobile carrier to transfer the victim's phone number to a device the attacker controls. One-time codes of all kinds can be captured by adversary-in-the-middle phishing kits, which relay the victim's interaction with the real sign-in page in real time and collect both the password and the code.

These techniques do not mean MFA is ineffective. They mean that the type of MFA matters. Phishing-resistant methods, such as (Fast IDentity Online) FIDO2 security keys and passkeys, bind authentication to the legitimate website and are considerably harder to relay or intercept. Chapter 4 examines these options and the practical difficulties of deploying them at enterprise scale.

Tokens and Sessions

An area that I believe many organizations still underestimate is the theft of tokens and session cookies. After a user authenticates, the identity provider or application typically issues a token or cookie that represents the authenticated session. For the lifetime of that token, the application generally trusts whoever presents it. If an attacker steals the token, they may be able to use the session without knowing the password and without triggering an MFA challenge at all.

4. Uber Technologies, Inc., "Security Update," *Uber Newsroom*, September 16, 2022 (updated September 19, 2022), <<https://www.uber.com/newsroom/security-update/>>, accessed October 22, 2026.

Adversary-in-the-middle phishing kits often capture session cookies directly. Information-stealing malware frequently harvests cookies from browsers on infected devices, including personal devices that employees use to access corporate SaaS applications. OAuth refresh tokens, which allow applications to obtain new access tokens without user interaction, can provide long-lived access if they are stolen or if a malicious application persuades a user to grant consent.

Token theft is particularly significant in the context of this book because AI agents and integrations rely heavily on tokens. An agent connected to email, documents, or business applications typically holds OAuth tokens or API keys that grant it ongoing access. Those tokens become attractive targets in their own right.

On-Premises Directory Attacks

In hybrid enterprises, the on-premises directory remains a rich source of identity-based attack techniques. Once an attacker has a foothold on a domain-joined machine, a well-documented set of methods can allow them to harvest credential material from memory, reuse password hashes without cracking them, request service tickets that can be attacked offline to recover service account passwords, and, if they reach sufficiently privileged accounts, forge authentication tickets that grant broad and persistent access to the domain. Many of these techniques have been publicly known for more than a decade. They remain effective in part because the underlying weaknesses, such as excessive administrative group membership, weak service account passwords, and administrators signing in to ordinary workstations, are organizational and architectural rather than simply technical.

ENTERPRISE FIELD NOTES

When I ask security teams how many accounts in their environment hold domain administrator or equivalent privileges, the first answer is usually a small number. When we actually enumerate nested group memberships, delegated permissions, and accounts with rights over critical directory objects, the true number is often several times larger. Effective privilege in a directory is rarely what the group named "Domain Admins" suggests. Attackers use tools specifically designed to map these hidden paths. Defenders should be at least as well informed about their own environment.

Identity as Initial Access and as a Path to Privilege

It is useful to think about identity's role in an attack at two distinct stages.

At the initial access stage, a compromised identity is the attacker's entry point. A phished employee credential, a stolen session token, a leaked cloud access key, or an abused third-party integration can each provide the first foothold. In environments that rely on SaaS and cloud services, initial access through identity may not require any interaction with the organization's own network or devices at all. The attacker simply signs in to the same services that employees use.

At the privilege escalation stage, identity becomes the means by which an attacker expands their reach. Having gained a foothold with a relatively low-privileged account, the attacker looks for ways to obtain more powerful access. Common paths include credentials stored in scripts, configuration files, or documentation; service accounts with

excessive permissions; misconfigured cloud IAM policies that allow an identity to grant itself additional rights; help desk processes that can be manipulated into resetting passwords or MFA devices for privileged users; and administrative accounts used on systems where their credentials can be harvested.

In cloud environments, privilege escalation often involves permissions that appear harmless in isolation. A role that can modify IAM policies, attach roles to compute resources, or create new access keys may be effectively equivalent to full administrative access, even if it was never labeled as such. In my experience, few organizations have a complete picture of these indirect escalation paths.

Social Engineering in the Age of AI

Social engineering has always targeted identity. The phishing email, the pretext phone call, and the impersonated executive are all attempts to persuade a person to surrender credentials, approve access, or take an action on the attacker's behalf. What AI changes is the quality, scale, and cost of these attacks.

Phishing messages produced with generative AI can be fluent, contextually appropriate, and tailored to the recipient's role, using information drawn from public profiles, company websites, and previous breaches. The spelling errors and awkward phrasing that security awareness programs taught employees to notice are becoming less reliable as warning signs. Attackers can generate large numbers of distinct, personalized messages, which may also make them harder for filtering systems to identify by pattern.

Voice cloning and video deepfakes extend impersonation beyond text. Publicly reported cases have described attackers using synthetic voice to impersonate executives in phone calls and, in at least one

widely reported incident, using deepfake video of multiple senior colleagues during a conference call to persuade an employee to authorize substantial financial transfers.⁵ The technology required to produce convincing synthetic media appears to be becoming more accessible, and I expect these techniques to become more common.

The help desk deserves special mention. Help desks exist to restore access for users who have lost it, which means they are, by design, empowered to reset passwords and MFA registrations. Attackers have repeatedly targeted this function, calling support staff while posing as employees, sometimes armed with personal details gathered from social media or prior breaches, and requesting a credential or MFA reset. Several significant publicly reported intrusions in recent years have been attributed to this technique.⁶ As voice cloning improves, verifying a caller's identity by voice or by knowledge of personal details is likely to become less reliable still.

Chapter 10 examines AI-enabled attacks in greater depth. The key point for now is that many organizations' identity verification processes, especially for account recovery and high-risk approvals,

5. Kathleen Magramo, "British Engineering Giant Arup Revealed as \$25 Million Deepfake Scam Victim," *CNN Business*, May 16, 2024, <<https://edition.cnn.com/2024/05/16/tech/arup-deepfake-scam-loss-hong-kong-intl-hnk>>, accessed October 22, 2026.

6. Federal Bureau of Investigation and Cybersecurity and Infrastructure Security Agency, *Scattered Spider*, Joint Cybersecurity Advisory AA23-320A, CISA, Washington, D.C., November 16, 2023 (updated July 29, 2025), <<https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a>>, accessed October 22, 2026.

were designed around assumptions about what is difficult to fake. Some of those assumptions may no longer hold.

BEST PRACTICES THAT WORK

Treat account recovery as a high-risk authentication event rather than a customer service transaction. In practice, this tends to mean requiring stronger verification for resets of privileged or sensitive accounts, such as an in-person or video check against a verified identity document, or confirmation from the user's manager through a separate, established channel. It also means alerting the user and the security team when MFA methods are changed, and treating a password reset followed quickly by an MFA registration from a new device as a signal worth investigating. The recovery process is frequently the weakest point in an otherwise strong authentication design.

Lateral Movement, Privilege Abuse, and Attack Chains

Real-world intrusions rarely depend on a single technique. They are chains of steps, each of which builds on the access obtained in the previous one. Identity is often the connective tissue that links those steps together.

Lateral movement is the process by which an attacker moves from one system or account to another within an environment. In traditional networks, this frequently involves reusing harvested credentials to sign in to additional machines. In cloud and SaaS environments, it may involve using one application's access to reach another, for example by using a compromised email account to reset passwords

for other services, or by using an integration's token to access data in a connected system. Federation and single sign-on, which provide enormous benefits for users and administrators, can also mean that compromising the central identity provider grants access to everything that trusts it.

Privilege abuse does not always involve an external attacker. Insiders, whether malicious or simply careless, can misuse legitimate access to view, copy, or modify data they should not. Excessive permissions magnify this risk. An employee who has accumulated access over years of role changes, a contractor whose access was never removed after their engagement ended, or an administrator with standing rights to sensitive systems all represent potential sources of harm that do not require any compromise at all.

Persistence is another important element of identity-based attack chains. Once attackers obtain privileged access, they frequently create mechanisms that allow them to return even if the original compromised account is discovered. These might include new user accounts, additional MFA methods registered to existing accounts, new OAuth applications with broad permissions, new access keys for cloud identities, or modifications to federation trust settings. Effective incident response, as discussed in Chapter 7, requires identifying and removing all of these footholds, which is considerably more difficult than simply resetting a single password.

REALISTIC THREAT SCENARIO

An attacker researches a mid-sized manufacturing company using professional networking sites and identifies several employees in the finance department. Using AI-generated content, the attacker sends a convincing message to one of them, referencing a real supplier and directing the recipient to a docu-

ment-sharing link. The link leads to an adversary-in-the-middle phishing page, which relays the employee's sign-in to the company's real identity provider, captures the password and one-time code, and steals the resulting session cookie.

Using the stolen session, the attacker signs in to the employee's email account and creates a mail rule that hides messages from the supplier. The attacker then searches the mailbox and shared drives, discovering a spreadsheet containing credentials for an older finance application and a document describing the payment approval process. The attacker also grants consent to a malicious OAuth application with mailbox read permissions, providing access that survives a later password reset.

Two weeks later, the attacker sends an email from the compromised account requesting a change to the supplier's banking details, followed by a short phone call to the accounts payable team using a cloned voice of the employee. A substantial payment is redirected before anyone notices.

At several points, this chain could have been interrupted: phishing-resistant authentication would likely have defeated the initial phishing page; conditional access policies binding sessions to managed devices might have prevented use of the stolen cookie; monitoring for new mail rules and OAuth consents could have provided early warning; removing stored credentials from shared drives would have closed an escalation path; and an out-of-band verification requirement for changes to payment details would have stopped the final step. No single control was decisive. Each one would have reduced the attacker's chance of success.

Non-Human Identities in the Attack Surface

Most discussions of identity-based attacks focus on human users, but non-human identities are an increasingly important part of the attack surface. Attackers search public code repositories, container images, and misconfigured storage for exposed API keys and cloud credentials, and automated tools can find and exploit such secrets quickly after they are published. Compromised CI/CD pipelines can expose the credentials used to deploy software into production. Third-party SaaS integrations hold tokens that grant access to customer data, and a breach at an integration provider can expose every customer that granted it access.

AI agents add further dimensions. An agent that processes untrusted content, such as incoming email, web pages, or documents, may be susceptible to prompt injection, in which instructions embedded in that content attempt to redirect the agent's behavior. If the agent holds broad permissions, a successful injection could cause it to retrieve sensitive data, send messages, or take actions on the attacker's behalf, all under a legitimate identity. In that scenario, the agent becomes an unwitting insider. Chapter 9 addresses how to constrain and monitor agents to limit this risk.

Mapping Identity Threats to Enterprise Risk and Business Impact

Understanding attack techniques is necessary, but it is not sufficient. Security leaders must also explain identity risk in terms that executives, boards, and business owners can use to make decisions. In my experience, identity programs that are framed primarily in technical

terms struggle to secure sustained funding, while programs connected clearly to business outcomes tend to fare better.

One useful approach is to begin with the organization's most important business processes and assets rather than with its identity systems. What are the critical operations, the most sensitive data, the most consequential financial flows, and the systems whose disruption would cause the greatest harm? For each of these, security teams can then ask: Which identities, human and non-human, have access? How many of them hold privileged or standing access? How strong is the authentication protecting them? How would we detect misuse? How quickly could we revoke access if needed?

This approach tends to reveal concentrations of risk that are not visible from an identity-centric inventory alone. A single service account with broad access to a payment system, an integration with read access to the entire customer database, or a small group of administrators whose accounts control the cloud environment may represent a disproportionate share of the organization's exposure.

Identity risk can also be expressed through scenarios. Rather than reporting that a certain percentage of accounts lack phishing-resistant MFA, a security leader might describe how a compromised account in a particular group could lead to a specific business impact, such as fraudulent payments, regulatory notification obligations following a data breach, or operational disruption from ransomware. Scenarios like the one presented earlier in this chapter can help decision-makers understand how individual weaknesses combine and why investments in seemingly unrelated controls reinforce one another.

Finally, I would encourage security leaders to be candid about uncertainty. Quantifying the likelihood and impact of identity-based attacks is genuinely difficult, and claims of precise financial risk figures may undermine credibility if they cannot be defended. Ranges, com-

parisons, and well-reasoned scenarios are often more persuasive than false precision.

PRACTITIONER CHECKLIST

Which accounts in your environment are exempt from MFA, and is each exemption documented, justified, owned, and time-limited?

What proportion of your users, and specifically your privileged users, rely on phishing-resistant authentication?

How long do session tokens and refresh tokens remain valid for your most sensitive applications, and can sessions be bound to managed devices?

How does your help desk verify identity before resetting passwords or MFA methods, especially for privileged accounts?

Do you monitor for new OAuth application consents, new mail forwarding rules, new MFA registrations, and changes to federation settings?

Have you mapped the effective privileged access in your directory and cloud environments, including nested groups and indirect escalation paths?

Do you scan code repositories, pipelines, and shared storage for exposed secrets?

Can you identify which identities have access to your most critical business processes and data?

Looking Ahead

The techniques described in this chapter share a common feature: they exploit the trust that systems place in identity. Attackers do not need to defeat that trust if they can borrow it. Defending against them, therefore, requires an architecture in which trust is established

carefully, granted narrowly, evaluated continuously, and monitored closely.

Chapter 3 turns to that architecture. It describes the core components of an enterprise identity security architecture, from identity providers and directories to federation and single sign-on; examines how to integrate identity security across cloud, on-premises, SaaS, and legacy systems; explains how Zero Trust principles translate into identity-aware access decisions; and discusses the architectural patterns and common design mistakes I have encountered in enterprise implementations.

3. Identity Security Architecture for the Modern Enterprise

Early in one engagement, I asked a client's identity team to draw their identity architecture on a whiteboard. Three people stepped up, and each drew something different. One drew the on-premises directory at the center, with applications radiating outward. Another placed the cloud identity provider at the center and treated the directory as a legacy feed. The third drew a cluster of boxes representing the HR system, the provisioning platform, and the governance tool, with authentication barely mentioned. None of them was wrong, exactly. Each was describing the part of the architecture they worked with every day. What none of them could draw was the

whole picture, including how the pieces depended on one another and where trust actually flowed.

That experience has stayed with me because I believe it is common. In most large enterprises, identity architecture is not designed so much as accumulated. It grows through acquisitions, platform migrations, vendor selections made by different teams at different times, and urgent projects that solved one problem while creating dependencies that nobody documented. The result often works well enough day to day, but it can be fragile, difficult to secure, and very difficult to change.

This chapter describes the core components of an enterprise identity security architecture and how they fit together. It examines identity providers, directories, federation, and single sign-on; discusses how to integrate identity security across cloud, on-premises, SaaS, and legacy environments; explains how Zero Trust principles translate into identity-aware access decisions; and closes with the architectural patterns and design mistakes I have encountered most often. My goal is not to prescribe a single reference architecture, since every enterprise's starting point is different, but to provide a framework for understanding what a mature architecture needs to accomplish and where the most significant risks tend to sit.

The Core Components

At a conceptual level, an enterprise identity security architecture needs to answer a series of questions every time an entity attempts to do something. Who or what is this? How confident are we in that claim? What is it entitled to do? Is the current context appropriate for this request? Was the decision recorded, and would we notice if something

looked wrong? Different components address different parts of that sequence. I find it helpful to group them into six functional layers.

Identity Sources of Truth

Every identity architecture depends on authoritative sources that establish who exists and what their relationship to the organization is. For employees, this is usually the human resources system. For contractors and partners, it may be a vendor management system or, too often, a spreadsheet. For customers, it is typically a customer identity platform. For non-human identities, authoritative sources are frequently missing altogether, a problem examined in depth in Chapter 8.

The quality of these sources matters more than many architects initially assume. If the HR system does not reliably record terminations, role changes, and leaves of absence, every downstream control inherits that inaccuracy. In several organizations I have worked with, the most effective identity security improvement in a given year was not a new technology but a cleanup of the data flowing from the HR system into the identity platform.

Directories and Identity Stores

Directories store identity records, attributes, group memberships, and, in many cases, credentials. In most large enterprises, the on-premises directory, most commonly Active Directory, remains deeply embedded, supporting domain-joined devices, file shares, and a long tail of applications that depend on Kerberos or LDAP. Alongside it, organizations typically operate one or more cloud directories associated with their primary identity provider or productivity suite,

as well as directories embedded within cloud platforms and individual applications.

Identity Providers and Authentication Services

The identity provider (IdP) authenticates users and issues the assertions or tokens that applications rely on. In a modern enterprise, the IdP is arguably the single most important security component in the environment. It enforces authentication strength, applies conditional access policies, and serves as the gateway to hundreds or thousands of federated applications. That centrality is precisely what makes it so valuable to defenders and so attractive to attackers.

Access Management and Policy Enforcement

Authentication establishes identity; authorization determines what that identity may do. Authorization decisions are made in many places: within the IdP through conditional access, within individual applications through roles and permissions, within cloud platforms through IAM policies, and increasingly through dedicated policy engines that evaluate requests against centrally defined rules. Privileged access management, discussed in Chapter 6, adds additional enforcement for the most sensitive operations.

Identity Governance and Lifecycle Management

Identity governance and administration (IGA) platforms manage the lifecycle of identities and entitlements: provisioning access when someone joins, adjusting it when their role changes, removing it when they leave, and periodically reviewing whether existing access remains

appropriate. Chapter 5 examines governance in detail. Architecturally, the key point is that governance depends on connectivity. An IGA platform can only govern the systems it is connected to, and in many enterprises a substantial share of applications are not connected at all.

Visibility, Detection, and Response

The final layer collects and analyzes identity telemetry: authentication events, token issuance, privilege changes, group modifications, consent grants, and configuration changes to identity infrastructure. This data feeds security operations, identity threat detection and response capabilities, and forensic investigations. I have seen many architectures that were thoughtfully designed in their first five layers and treated this sixth layer as an afterthought. As Chapter 2 illustrated, an attacker using valid credentials may be visible only through identity telemetry, so this layer deserves the same design attention as the others.

ENTERPRISE FIELD NOTES

One exercise I have found consistently valuable is asking a simple question about every major application: if an employee were terminated at 9:00 this morning, at what time would their access to this application actually end? For applications federated through the primary IdP with automated deprovisioning, the answer is often within minutes. For applications with local accounts, manual provisioning, or long-lived sessions, the answer may be days, weeks, or never. Mapping those answers across the application portfolio tends to reveal more about the real state of an identity architecture than any diagram.

Directories, Federation, and Single Sign-On

Federation and single sign-on are the mechanisms that allow a central identity provider to extend its authority across many applications. Instead of each application maintaining its own credentials, the application trusts the IdP to authenticate the user and accepts a signed assertion or token as proof. SAML 2.0 remains widely used for enterprise web applications. OpenID Connect, built on OAuth 2.0, has become the prevailing standard for modern web and mobile applications. Provisioning standards such as SCIM allow the IdP or governance platform to create, update, and remove accounts in connected applications automatically.¹

The security benefits of this model are substantial. Users manage one strong credential rather than dozens of weak ones. Authentication policies, including MFA and conditional access, are enforced consistently in one place. Deprovisioning at the IdP can cut off access to every federated application at once. Authentication events are logged centrally, which greatly improves visibility.

Federation also concentrates trust, and that concentration carries risk. If an attacker compromises the IdP, or obtains the ability to modify its federation settings or signing keys, they may be able to impersonate any user to any application that trusts it. Publicly report-

1. Phil Hunt, Kelly Grizzle, Morteza Ansari, Erik Wahlstroem, and Chuck Mortimore, *System for Cross-domain Identity Management: Protocol*, RFC 7644, Internet Engineering Task Force, September 2015, <<https://www.rfc-editor.org/rfc/rfc7644>>, accessed October 22, 2026.

ed incidents have shown attackers forging authentication assertions after compromising federation infrastructure,² and adding their own federation trusts to cloud tenants so that identities they control are accepted as legitimate.³ These are among the most severe identity compromises an organization can experience, because they undermine the foundation that other controls depend on.

Single sign-on also tends to extend the value of a single session. A user who authenticates once may gain access to many applications for the lifetime of that session. If the session is hijacked, the attacker inherits that breadth. Session lifetimes, re-authentication requirements for sensitive applications, and the ability to revoke sessions centrally are therefore important architectural decisions rather than minor configuration details.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Single identity provider or multiple? Consolidating on a single enterprise IdP simplifies policy enforcement, reduces administrative

2. Cybersecurity and Infrastructure Security Agency, *Detecting Post-Compromise Threat Activity in Microsoft Cloud Environments*, Alert AA21-008A, CISA, Washington, D.C., January 8, 2021, <<https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-008a>>, accessed October 22, 2026.

3. Microsoft Threat Intelligence, Microsoft Defender Experts, and Veronica Sun, "Octo Tempest Crosses Boundaries to Facilitate Extortion, Encryption, and Destruction," *Microsoft Security Blog*, October 25, 2023, <<https://www.microsoft.com/en-us/security/blog/2023/10/25/octo-tempest-crosses-boundaries-to-facilitate-extortion-encryption-and-destruction/>>, accessed October 22, 2026.

overhead, and improves visibility. It also creates a single point of failure and a single high-value target, and it may be difficult to achieve in organizations with significant acquisitions, regional regulatory requirements, or separate operational technology environments.

Maintaining multiple IdPs can provide resilience and accommodate organizational boundaries, but it tends to produce inconsistent policies, duplicated identities, and gaps between systems that attackers can exploit.

In my experience, most large enterprises benefit from designating one primary workforce IdP as the authoritative point of authentication and policy enforcement, federating any remaining IdPs to it or planning their consolidation, and investing heavily in protecting and monitoring that primary IdP as tier-zero infrastructure. A documented break-glass procedure for IdP outages is essential under either model.

Integrating Across Cloud, On-Premises, SaaS, and Legacy Systems

If every application supported modern federation, provisioning, and fine-grained authorization, identity architecture would be considerably simpler. In practice, enterprise application portfolios span several generations of technology, and integration is where many identity programs encounter their greatest difficulties.

Hybrid Directory Synchronization

Most enterprises that use a cloud IdP continue to synchronize identities from the on-premises directory. This synchronization creates an important trust relationship that architects sometimes overlook. If the

synchronization infrastructure or the on-premises directory is compromised, an attacker may be able to modify identities that flow into the cloud, or, depending on configuration, move from on-premises compromise to cloud compromise. Conversely, some configurations allow cloud changes to write back to the on-premises directory. Understanding the direction of trust, protecting synchronization servers as tier-zero assets, and keeping highly privileged cloud administrative accounts as cloud-only identities rather than synchronized ones are among the practices I recommend most often.

Cloud Platform Identity

Each major cloud platform has its own identity and access management model, with its own concepts of users, roles, policies, and service identities. Workforce access to cloud consoles and APIs should generally flow through federation from the enterprise IdP rather than through local cloud user accounts. Workloads running within the cloud should use the platform's native workload identity mechanisms where possible, rather than long-lived access keys. Multi-cloud environments multiply the complexity, since permissions models differ significantly across platforms and few tools provide a complete, consistent view of effective access across all of them.

SaaS Applications

SaaS adoption has made federation more common, but it has also introduced new gaps. Not every SaaS application supports SSO, and some reserve it for premium licensing tiers. Not every application supports automated provisioning. Many SaaS applications have their own administrative roles, API tokens, and third-party integrations that sit

outside the IdP's view. Shadow SaaS, meaning applications adopted by business units or individuals without central review, adds identities and data flows that the security team may not know exist. Discovering SaaS usage and bringing high-risk applications under central identity control is, in my experience, an ongoing effort rather than a one-time project.

Legacy Applications

Legacy applications present the most stubborn challenges. Some authenticate against their own internal user tables. Some rely on the on-premises directory using older protocols. Some cannot support MFA at all. Some are business-critical, poorly documented, and maintained by a shrinking number of people who understand them.

There is rarely a single answer for legacy systems. Options include placing them behind an identity-aware proxy or application gateway that enforces modern authentication before traffic reaches the application; restricting network access so they can be reached only from managed devices or privileged access workstations; wrapping administrative access through a privileged access management platform; compensating with enhanced monitoring; and, ultimately, retiring or replacing them. The practical approach is usually to prioritize by risk, focusing first on legacy systems that hold sensitive data or support critical processes.

BEST PRACTICES THAT WORK

Build and maintain an application identity inventory that records, for each significant application, how users authenticate, whether it is federated to the primary IdP, whether provisioning is automated, whether it is connected to governance processes, who owns it, what data it holds, and whether it has local or emergency accounts. This

inventory sounds mundane, and it is. It is also the foundation for nearly every prioritization decision in an identity security program. Organizations that maintain it well can answer questions about exposure in hours. Organizations that do not may need weeks.

Zero Trust and Identity-Aware Access Decisions

Zero Trust has become one of the most widely used terms in cybersecurity, and also one of the most diluted. Vendors apply the label to a wide range of products, and executives sometimes speak of Zero Trust as though it were something an organization could purchase and install. In my view, it is more useful to treat Zero Trust as a set of design principles that shape how access decisions are made.

NIST Special Publication 800-207 describes several core tenets, which I would summarize for the purposes of this book as follows.⁴ No network location confers trust by itself. Access to each resource is granted per session, based on policy. Policy draws on multiple signals, including the identity of the requester, the state of the device, the sensitivity of the resource, and behavioral or environmental context. Access is granted with the least privilege necessary. The organization continuously monitors the security posture of its assets and uses that information to improve decisions.

NIST's model also introduces a helpful architectural vocabulary. A policy decision point evaluates requests against policy and determines whether access should be granted. A policy enforcement point sits in the path of the request and carries out that decision. Signals from identity systems, device management, threat intelligence, and activ-

4. Rose et al., *Zero Trust Architecture*, NIST SP 800-207, sec. 2.1.

ity logs inform the decision. In a typical enterprise implementation, the IdP's conditional access engine acts as a policy decision point for authentication, while identity-aware proxies, application gateways, cloud platform controls, and the applications themselves serve as enforcement points.

Identity-aware access decisions, in practice, combine several categories of signal. Identity signals include who the user or workload is, which groups or roles it holds, and the strength of the authentication method used. Device signals include whether the device is managed, whether it meets security baselines, and whether endpoint protection reports any concerns. Context signals include location, time, network, and the sensitivity of the requested resource. Risk signals include indications from the IdP or detection tools that an account or session may be compromised, such as sign-ins from unfamiliar infrastructure or credentials found in a known leak.

The most mature implementations I have seen evaluate these signals not only at sign-in but continuously during a session. If a device falls out of compliance, if a user's risk level rises, or if an administrator revokes access, sessions are re-evaluated or terminated rather than being allowed to continue until expiry. Standards and vendor capabilities in this area are still developing, and support across applications is uneven, but I believe continuous evaluation will become increasingly important as AI agents operate long-running sessions on behalf of users.

A word of caution is warranted. Zero Trust programs can fail when they are pursued as a sweeping transformation with a vague end state. The programs I have seen succeed tended to define concrete, incremental objectives, such as requiring managed devices for access to a specific class of sensitive applications or eliminating standing access to cloud administrative roles, and then expanding from there.

Architectural Patterns and Common Design Mistakes

Over many engagements, I have seen certain patterns recur, some productive and some problematic. The following observations are not exhaustive, but they reflect the issues I encounter most often.

Patterns That Tend to Work

Tiered administration separates the accounts and workstations used to manage the most sensitive infrastructure, such as identity systems and domain controllers, from those used for ordinary work.⁵ Administrators use dedicated privileged accounts on hardened devices for tier-zero tasks, and those credentials are never exposed on lower-tier systems where they could be harvested. This model takes discipline to maintain, but it substantially limits the escalation paths described in Chapter 2.

Centralized policy with distributed enforcement allows the organization to define access policy once, in the IdP or a policy engine, and apply it consistently through many enforcement points. It reduces the risk that different applications implement security requirements inconsistently.

5. Microsoft Corporation, "Privileged Access Devices," *Microsoft Learn*, <<https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-devices>>, accessed October 22, 2026.

Separation of identity planes means keeping the identities that administer the identity system itself distinct from the identities they administer. Cloud-only emergency administrator accounts, protected by phishing-resistant authentication and stored securely, provide a path to recovery if the primary authentication flow or synchronization fails or is compromised.

Identity telemetry by design means that logging requirements are defined at the same time as integration requirements. When a new application is onboarded, the team confirms not only that users can sign in but that authentication events, administrative actions, and permission changes are captured and forwarded to security operations.

Mistakes I See Repeatedly

The first and most common mistake is treating the IdP as an IT utility rather than as critical security infrastructure. Administrative roles in the IdP are granted too broadly, changes are not monitored, and break-glass accounts are created but not protected or tested.

The second is assuming that federation equals protection. An application federated to the IdP may still maintain local accounts that bypass SSO, API tokens that never expire, or administrative backdoors created during implementation. Federation addresses the front door. It does not necessarily close the side entrances.

The third is synchronizing everything everywhere. Synchronizing all on-premises accounts and groups into the cloud, including privileged on-premises accounts, can create pathways from one environment's compromise into the other. Synchronization scope deserves deliberate design.

The fourth is designing for human identities only. Architectures built around workforce authentication often have no coherent answer for service accounts, workloads, API integrations, or AI agents. These identities end up managed through ad hoc processes, embedded secrets, and local credentials, which becomes a major source of risk as their numbers grow.

The fifth is underinvesting in resilience. When the IdP becomes the gateway to everything, an outage or a destructive attack against it can halt the business. Recovery plans for identity infrastructure, including the ability to restore the on-premises directory from a trusted backup and to regain administrative control of cloud tenants, are frequently untested. In my experience, organizations tend to discover these gaps during an incident, which is the worst possible time.

The sixth is pursuing tool consolidation as an end in itself. Reducing the number of identity products can simplify operations, but replacing one platform with another without addressing data quality, ownership, and process issues often reproduces the same problems on new infrastructure.

REALISTIC THREAT SCENARIO

An attacker compromises a developer's workstation through a malicious software package and harvests credentials cached on the device. Among them is the password for a service account used by the directory synchronization server, which had been granted broad permissions years earlier during an upgrade and never reduced. Using that account, the attacker reaches the synchronization server, which had not been treated as tier-zero infrastructure and was administered by the same team and accounts as ordinary application servers.

From the synchronization server, the attacker is able to influence identities flowing into the cloud tenant and to obtain access to a synchronized on-premises account that had been granted a highly

privileged cloud administrative role for convenience. With that role, the attacker adds a new federation trust to the cloud tenant, allowing them to authenticate as any user without needing additional credentials.

The controls that would most likely have prevented this chain were architectural rather than product-specific: treating synchronization infrastructure as tier-zero, keeping privileged cloud roles assigned only to cloud-only accounts protected with phishing-resistant authentication, minimizing service account permissions, and alerting on any change to federation configuration. Detection of the final step, a new federation trust, would likely have been the last practical opportunity to contain the incident before it became enterprise-wide.

PRACTITIONER CHECKLIST

Can your team produce an accurate, current diagram of your identity architecture, including sources of truth, directories, IdPs, synchronization flows, and federation trusts?

Is your primary IdP, along with its synchronization and federation infrastructure, administered and monitored as tier-zero?

Are highly privileged cloud administrative roles held only by cloud-only accounts with phishing-resistant authentication?

What proportion of your applications are federated to the primary IdP, and how many still maintain local accounts or non-expiring tokens?

Which legacy applications cannot support modern authentication, and what compensating controls protect them?

Do your access policies incorporate device state and risk signals, and can sessions be revoked centrally?

Is there a tested recovery plan for loss or compromise of your directory and IdP?

Does your architecture include a defined approach for non-human identities and AI agents?

Looking Ahead

A sound architecture provides the structure within which identity security operates, but architecture alone does not determine outcomes. The strength of each control within that structure matters a great deal, and no control matters more than authentication, the moment at which an identity claim is first accepted.

Part II of this book turns to the core capabilities that build on this architectural foundation. Chapter 4 begins with authentication: the strengths and weaknesses of passwords, MFA, biometrics, and passkeys; why phishing-resistant authentication has become essential; how federation protocols and adaptive authentication fit together; and how organizations can balance security, usability, resilience, and operational complexity as they modernize the way identities prove who they are.

4. Authentication and the Future of Digital Trust

A few years ago, I was part of a team that rolled out a new multifactor authentication requirement across a large global workforce. On paper, the project was a success. Within a few months, nearly every employee was enrolled, the dashboards were green, and leadership was satisfied that a major risk had been addressed. Then the security operations team began noticing something in the logs: a small but steady stream of push notifications that users had approved even though the sign-in attempts came from locations and devices the users had never used. When we followed up, some employees told us they had approved the prompts because they assumed IT was testing something. Others said the notifications arrived so frequently during the day that approving them had become automatic.

That experience taught me a lesson I have carried into every authentication program since. Deploying MFA is not the same as establishing strong authentication. The method matters, the user experience matters, the recovery process matters, and the attacker's ability

to adapt matters. Authentication is the moment at which an identity claim is first accepted, and nearly every other control in this book depends on that moment being trustworthy.

This chapter examines authentication in depth. It reviews the fundamentals of passwords, MFA, biometrics, and passkeys; explains why phishing-resistant authentication has become essential and where traditional MFA falls short; describes how single sign-on, federation, and modern identity protocols carry authentication across the enterprise; discusses adaptive and risk-based authentication; and closes with the practical trade-offs between security, usability, resilience, and operational complexity that determine whether an authentication program succeeds in the real world.

Authentication Fundamentals

Authentication is traditionally described in terms of factors: something you know, such as a password; something you have, such as a phone or security key; and something you are, such as a fingerprint or face. Multifactor authentication combines two or more of these categories so that compromising a single factor is not sufficient to impersonate the user. The framework remains useful, but I have found that it can also be misleading if it is treated as a checklist. Two factors that are both vulnerable to the same attack, such as a password and a one-time code that are both entered into the same phishing page, provide less protection than the label "multifactor" suggests.

Passwords

Passwords have been declared obsolete many times, and they persist in almost every enterprise environment I have seen. They are familiar,

universally supported, and inexpensive to implement. They are also vulnerable to phishing, reuse, guessing, and theft, and they impose a real burden on users and help desks.

Guidance on password policy has changed significantly over the past decade. NIST's digital identity guidelines, in Special Publication 800-63B and its subsequent revisions, moved away from many practices that organizations had long considered standard.¹ The guidance favors longer passwords and passphrases over complex character composition rules, recommends screening new passwords against lists of commonly used and previously breached passwords, and discourages mandatory periodic password changes in the absence of evidence of compromise. In my experience, many enterprise password policies still reflect older practices, often because they are embedded in audit requirements or legacy system configurations. Updating them can reduce user friction and help desk volume without weakening security, and in some cases it may strengthen it.

Even a well-designed password policy, however, does not address the core weakness: a password is a shared secret that the user can be tricked into revealing. For that reason, I regard passwords as a factor to be supplemented and, over time, reduced or removed, rather than one to be perfected.

1. David Temoshok, James L. Fenton, Yee-Yin Choong, Naomi Lefkovitz, Andrew Regenscheid, Ryan Galluzzo, and Justin P. Richer, *Digital Identity Guidelines: Authentication and Authenticator Management*, NIST Special Publication 800-63B-4, National Institute of Standards and Technology, Gaithersburg, Md., July 2025, <<https://doi.org/10.6028/NIST.SP.800-63b-4>>, accessed October 22, 2026.

Traditional Multifactor Authentication

Most enterprise MFA deployments rely on one or more of the following: one-time codes sent by SMS or voice call, time-based one-time passwords generated by an authenticator app or hardware token, and push notifications sent to a mobile app for approval. Each represents a substantial improvement over passwords alone, and organizations that have not yet deployed MFA broadly should treat doing so as an urgent priority.

Each method also has known weaknesses, several of which Chapter 2 described. SMS and voice codes are exposed to SIM swapping and interception. Codes of all kinds can be captured and relayed by adversary-in-the-middle phishing pages. Push notifications can be abused through fatigue attacks. Vendors have introduced mitigations such as number matching, which requires the user to enter a number displayed on the sign-in screen into the app, and additional context showing the location and application of the sign-in request.² These mitigations help considerably against fatigue attacks. They do not, in general, defeat a real-time phishing proxy, because the user is interacting with what appears to be a legitimate sign-in and will see a legitimate number to enter.

2. Cybersecurity and Infrastructure Security Agency, *Implementing Number Matching in MFA Applications*, Fact Sheet, CISA, Washington, D.C., October 2022, <<https://www.cisa.gov/sites/default/files/publications/fact-sheet-implement-number-matching-in-mfa-applications-508c.pdf>>, accessed October 22, 2026.

Biometrics

Biometric authentication, typically fingerprint or facial recognition, is now built into most laptops and mobile devices. In well-designed enterprise implementations, biometrics are used locally to unlock a credential stored securely on the device, rather than being transmitted to a server and compared against a central database. This distinction matters. When the biometric match happens on the device and unlocks a cryptographic key, the biometric data itself is not exposed to network attackers, and compromise of a server does not reveal users' fingerprints or facial templates.

Centralized biometric systems, in which biometric data is stored and matched by a service, raise more significant privacy, regulatory, and security concerns, since biometric characteristics cannot be changed if they are compromised. Organizations should also be aware that biometric verification, particularly facial and voice recognition used in remote identity proofing, is a target for the deepfake and synthetic media techniques discussed in Chapter 10.

Passkeys and FIDO2

The most significant development in authentication in recent years, in my view, has been the maturation of FIDO2 and the broad adoption of passkeys. FIDO2 combines the W3C Web Authentication standard, known as WebAuthn, with the FIDO Alliance's Client to

Authenticator Protocol.³ Together, they allow a user to authenticate using a cryptographic key pair. The private key never leaves the user's authenticator, which may be a hardware security key, a platform authenticator built into a laptop or phone, or a credential manager. The server stores only the public key.

When the user signs in, the server sends a challenge, and the authenticator signs it with the private key after verifying the user locally, usually with a biometric or PIN. Critically, the signature is bound to the origin of the website requesting it. A phishing site at a look-alike domain cannot obtain a valid signature for the real site, because the authenticator will not produce one for the wrong origin. This property is what makes FIDO2 phishing-resistant rather than merely stronger.

Passkeys are FIDO credentials designed for broad consumer and enterprise usability. They come in two general forms.⁴ Device-bound passkeys remain on a single authenticator, such as a hardware security key or a specific device's secure hardware. Synced passkeys are stored in a credential manager and synchronized across a user's devices, which

3. World Wide Web Consortium, *Web Authentication: An API for Accessing Public Key Credentials, Level 3*, W3C Recommendation, W3C, August 25, 2026, <<https://www.w3.org/TR/webauthn-3/>>, accessed October 22, 2026.

4. FIDO Alliance Enterprise Deployment Working Group, Dean H. Saxe, ed., *FIDO Deploying Passkeys in the Enterprise: Introduction*, White Paper, FIDO Alliance, August 29, 2024, <<https://fidoalliance.org/white-paper-fido-deploying-passkeys-in-the-enterprise-introduction/>>, accessed October 22, 2026.

greatly improves convenience and recovery but means the security of the passkey depends partly on the security of the synchronization service and the account that controls it. For most workforce users, I believe either form represents a substantial improvement. For highly privileged administrators, many organizations reasonably prefer device-bound credentials, often on dedicated hardware security keys, so that they retain tighter control over where those credentials exist.

ENTERPRISE FIELD NOTES

In passkey and security key rollouts I have been involved with, the technology itself was rarely the main obstacle. The difficulties were operational: issuing and replacing hardware keys across many countries, handling employees who lost or forgot their devices, supporting shared workstations in plants and call centers, accommodating applications that did not support modern authentication, and redesigning help desk procedures so that account recovery did not quietly reintroduce the weaker methods the program was meant to eliminate. Programs that planned for these issues from the start tended to move faster than those that discovered them during deployment.

Phishing-Resistant Authentication and the Limits of Traditional MFA

The case for phishing-resistant authentication rests on a simple observation. Most traditional MFA methods depend on the user correctly identifying whether they are interacting with a legitimate site or service. Attackers have become very good at defeating that judgment, and, as Chapter 2 discussed, AI is likely to make them better still.

Phishing-resistant methods remove that judgment from the user and place it in the cryptographic protocol.

In practice, two families of methods are generally regarded as phishing-resistant: FIDO2 and passkeys, and certificate-based authentication using smart cards or similar mechanisms such as the PIV and CAC cards used in the U.S. government. Both rely on cryptographic keys that are bound either to the origin being authenticated to or to a mutually authenticated channel. U.S. federal guidance has required agencies to adopt phishing-resistant MFA for their workforces,⁵ and CISA has encouraged critical infrastructure and private sector organizations to prioritize it, particularly for privileged accounts and high-value targets.⁶

It is important to be clear about what phishing-resistant authentication does and does not accomplish. It substantially reduces the risk of credential phishing and real-time relay attacks against the authentication step itself. It does not, by itself, prevent the theft of session tokens from a compromised device after authentication has occurred.

5. Shalanda D. Young, *Moving the U.S. Government Toward Zero Trust Cybersecurity Principles*, Memorandum M-22-09, Office of Management and Budget, Washington, D.C., January 26, 2022, <<https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf>>, accessed October 22, 2026.

6. Cybersecurity and Infrastructure Security Agency, *Implementing Phishing-Resistant MFA*, Fact Sheet, CISA, Washington, D.C., October 2022, <<https://www.cisa.gov/sites/default/files/publications/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf>>, accessed October 22, 2026.

It does not prevent a user from being socially engineered into taking a harmful action while legitimately signed in. And it does not protect an account if the attacker can bypass it altogether, for example through a weak account recovery process or a legacy protocol that still accepts passwords alone.

That last point deserves emphasis. I have seen organizations deploy phishing-resistant authentication for their main sign-in flow while leaving older authentication protocols enabled for compatibility, or while allowing users to fall back to SMS codes if they did not have their security key with them. In each case, an attacker who understood the environment could simply choose the weaker path. A phishing-resistant program is only as strong as the weakest method the system will still accept for a given account.

BEST PRACTICES THAT WORK

Prioritize phishing-resistant authentication by risk rather than attempting to convert the entire workforce at once. In most enterprises, a sensible sequence begins with administrators of identity infrastructure and cloud platforms, followed by other privileged users, then executives and their assistants, finance and payment roles, help desk staff, and developers with access to production systems, before expanding to the general workforce. At each stage, disable or restrict the weaker methods for that population once enrollment is complete, enforce the requirement through conditional access policy, and monitor for attempts to use fallback methods. Enrollment without enforcement leaves the old attack paths open.

Single Sign-On, Federation, and Modern Identity Protocols

Chapter 3 described federation and SSO from an architectural perspective. From an authentication perspective, their significance is that they allow strong authentication to be performed once, at the identity provider, and then conveyed to many applications in the form of signed assertions or tokens. This is how an organization can enforce phishing-resistant authentication for hundreds of applications without each application needing to implement it.

The protocols involved each play a distinct role. SAML 2.0 conveys signed XML assertions about an authenticated user to a service provider, and it remains common in enterprise web applications. OAuth 2.0 is a framework for delegated authorization, allowing a client application to obtain an access token that grants specific, scoped permissions on a user's behalf. OpenID Connect builds an authentication layer on top of OAuth 2.0, providing an ID token that tells the application who the user is and how they authenticated. Ongoing work in the standards community, including the consolidation of OAuth best practices in the draft OAuth 2.1 specification, aims to remove older flows that proved insecure and to make safer defaults standard.⁷

7. Dick Hardt, Aaron Parecki, and Torsten Lodderstedt, *The OAuth 2.1 Authorization Framework*, Internet-Draft draft-ietf-oauth-v2-1-16, Internet Engineering Task Force, September 3, 2026, <<https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/>>, accessed October 22, 2026.

Several aspects of these protocols matter particularly for security teams. First, assertions and tokens can carry information about how the user authenticated, often expressed as an authentication context or authentication methods reference. Applications that handle sensitive operations can use this information to require that the user authenticated with a phishing-resistant method, or recently, before allowing certain actions. Second, token lifetimes and refresh behavior directly determine how long a stolen token remains useful. Third, some OAuth flows designed for convenience, such as the device authorization flow used to sign in on devices without keyboards, have been abused in phishing campaigns that trick users into authorizing an attacker's session.⁸ Organizations should restrict such flows to the scenarios that genuinely need them.

A promising area of development is binding tokens and sessions more tightly to the device on which they were issued, so that a stolen token cannot simply be replayed from another machine. Approaches such as proof-of-possession tokens, including the OAuth DPoP

8. Microsoft Threat Intelligence, "Storm-2372 Conducts Device Code Phishing Campaign," *Microsoft Security Blog*, February 13, 2025, <<https://www.microsoft.com/en-us/security/blog/2025/02/13/storm-2372-conducts-device-code-phishing-campaign/>>, accessed October 22, 2026.

specification,⁹ and emerging browser mechanisms for device-bound session credentials are designed to address exactly this problem.¹

⁰ Support remains uneven across platforms and applications at the time of writing, but I expect these capabilities to become increasingly important, both for human sessions and for the long-lived tokens held by AI agents.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Session lifetime: long or short? Long sessions reduce authentication prompts and improve user experience, which in turn may reduce the habituation that makes users approve prompts carelessly. They also extend the window during which a stolen session token remains useful.

Short sessions limit that window but increase friction and can generate the kind of prompt fatigue that undermines security.

In my experience, the more effective approach is to vary session policy by risk. Ordinary applications accessed from managed, compliant devices can tolerate longer sessions, particularly where continuous evaluation can revoke them if risk changes. Privileged access, sensitive

9. Daniel Fett, Brian Campbell, John Bradley, Torsten Lodderstedt, Michael B. Jones, and David Waite, *OAuth 2.0 Demonstrating Proof of Possession (DPoP)*, RFC 9449, Internet Engineering Task Force, September 2023, <<https://www.rfc-editor.org/rfc/rfc9449>>, accessed October 22, 2026.

10. Mirko Zorz, "To Counter Cookie Theft, Chrome Ships Device-Bound Session Credentials," *Help Net Security*, April 10, 2026, <<https://www.helpnetsecurity.com/2026/04/10/google-chrome-device-bound-session-credentials/>>, accessed October 22, 2026.

data, and high-impact transactions warrant shorter sessions, step-up authentication at the moment of the sensitive action, and, where supported, device-bound tokens.

Adaptive Authentication, Device Context, and Risk-Based Access

Static authentication policies treat every sign-in the same way. Adaptive authentication adjusts requirements according to context and risk, asking for stronger proof when circumstances warrant it and reducing friction when they do not.

Device context is often the most valuable signal. Knowing whether a sign-in originates from a device the organization manages, whether that device meets security baselines such as disk encryption and current patches, and whether endpoint protection is running and reporting no threats allows the organization to make much better decisions than identity alone permits. Many enterprises now require managed, compliant devices for access to sensitive applications, which can also significantly reduce the value of stolen credentials and session tokens, since the attacker's device will not satisfy the policy.

Other signals include the network and location of the request, the time of access, the sensitivity of the application, and indications of risk from the identity provider or threat detection tools, such as sign-ins from anonymizing infrastructure, impossible travel between locations, or credentials appearing in known leaks. Risk-based policies can respond to these signals by requiring step-up authentication, limiting access to read-only functions, blocking access, or triggering an investigation.

I would offer two cautions about risk-based authentication. First, risk engines produce probabilistic assessments, not certainties. They

generate false positives that frustrate legitimate users and false negatives that allow attackers through, and they should be treated as one layer among several rather than as a substitute for strong authentication. Second, location-based signals have become less reliable as attackers increasingly route traffic through residential proxies and infrastructure located near their targets. Device-based and cryptographic signals tend to be harder for attackers to imitate.

Increasingly, authentication is also becoming continuous rather than a single event at sign-in. Standards such as the OpenID Foundation's Shared Signals Framework and the Continuous Access Evaluation Profile allow identity providers, applications, and security tools to exchange events about changes in risk, such as a revoked account, a compromised device, or a changed credential, so that sessions can be re-evaluated in near real time.¹¹ Adoption is still developing, but I believe this direction is the right one, particularly as AI agents begin operating long-lived sessions on behalf of users.

Balancing Security, Usability, Resilience, and Operational Complexity

Every authentication decision involves trade-offs, and in my experience, the programs that fail usually do so not because the chosen

11. "OpenID Foundation Sets New Standards for Real-Time Security Event Sharing," *Help Net Security*, September 22, 2025, <<https://www.helpnetsecurity.com/2025/09/22/openid-standards-real-time-security-event-sharing/>>, accessed October 22, 2026.

technology was weak but because these trade-offs were not managed well.

Usability

Authentication that is difficult to use tends to be circumvented. Users share credentials, approve prompts without reading them, store passwords insecurely, and pressure IT for exceptions. Fortunately, some of the strongest methods are also among the most convenient. Passkeys unlocked with a fingerprint or face are often faster than typing a password and entering a code. Organizations that position phishing-resistant authentication as an improvement in user experience, rather than as an additional burden, tend to see better adoption.

Resilience

As authentication becomes stronger and more centralized, the consequences of failure grow. If the identity provider is unavailable, or a critical population cannot authenticate because their devices are lost or a hardware supply chain is disrupted, business operations can halt. Resilient authentication design includes multiple registered authenticators per user, particularly for privileged users; secure, tested break-glass accounts; documented procedures for issuing temporary credentials during outages; and capacity planning for the help desk during large rollouts.

Account Recovery

Recovery deserves its own mention because it is so often the weak point. When a user loses their authenticator, the organization must

verify their identity and issue a new credential. If that process relies on knowledge-based questions, a phone call with minimal verification, or fallback to SMS, it may effectively reduce the account's security to the level of that process. Strong recovery approaches include requiring a second registered authenticator, verification through a manager or trusted colleague using an established channel, in-person verification for privileged accounts, and remote identity proofing services that check government identity documents. As Chapter 10 discusses, remote identity proofing is itself becoming a target for deepfakes, so organizations should evaluate such services carefully and monitor their effectiveness.

Operational Complexity

Large enterprises rarely have a uniform user population. Frontline workers on shared devices, contractors on unmanaged equipment, employees in regions with limited connectivity or restrictions on certain devices, and staff using legacy systems all require different approaches. A realistic authentication strategy acknowledges this diversity, defines a target state for each population, and accepts that some populations will move more slowly than others, while ensuring that compensating controls and documented exceptions cover the gaps in the meantime.

REALISTIC THREAT SCENARIO

A large services company completes a successful passkey roll-out for its workforce and enforces phishing-resistant authentication for access to email and collaboration tools. However, to minimize disruption, it leaves SMS codes enabled as a recovery

method and does not apply the new policy to a legacy remote access gateway that still accepts passwords and a one-time code.

An attacker targets a regional IT administrator, first through a phishing campaign that fails because the administrator's passkey will not authenticate to the lookalike site. The attacker then pivots. Using personal information gathered online, they persuade the administrator's mobile carrier to transfer the phone number to a new SIM. With control of the number, they initiate an account recovery, receive the SMS code, and register a new authentication method. Alternatively, they could have used a password obtained from an earlier breach together with an intercepted code to reach the legacy gateway.

The phishing-resistant deployment worked exactly as designed. The weakness was everything around it: a recovery method that undermined the primary method, and a legacy entry point excluded from policy. Removing SMS as a recovery option for administrators, requiring in-person or manager-verified recovery for privileged accounts, and bringing the legacy gateway behind the identity provider would likely have closed both paths.

PRACTITIONER CHECKLIST

What authentication methods are currently permitted in your environment, and for which populations? Which of them are phishing-resistant?

Have you defined a prioritized sequence for moving high-risk populations to phishing-resistant authentication, and are weaker methods disabled once each population is enrolled?

Are legacy authentication protocols disabled wherever possible, and are remaining exceptions documented and monitored?

Does your password policy reflect current guidance, including breached password screening and the removal of unnecessary periodic changes?

Do sensitive applications check how and when the user authenticated before allowing high-impact actions?

Are access policies informed by device compliance and risk signals, and can sessions be revoked when risk changes?

How does account recovery work for privileged users, and could it be used to bypass your strongest authentication methods?

Do privileged users have more than one registered authenticator, and are break-glass procedures documented and tested?

Looking Ahead

Authentication answers the question of who or what is making a request. It does not answer the equally important question of what that identity should be allowed to do. In my experience, many organizations have made meaningful progress on authentication in recent years while their authorization practices, including the accumulation of excessive permissions and the persistence of standing access, have lagged behind.

Chapter 5 addresses that gap. It examines the distinction between authentication and authorization, compares role-based and attribute-based access control, explores least privilege, separation of duties, and just-in-time access, and describes how identity governance and administration can help organizations reduce access sprawl and entitlement risk across complex enterprise environments.

5. Authorization, Least Privilege, and Access Governance

I once reviewed the access of a long-tenured employee at a large financial services firm as part of an internal investigation. Nothing malicious had happened; the investigation concerned a data handling error, and the employee's access was simply part of the background. What we found was striking nonetheless. Over roughly fifteen years, the employee had held seven different roles across three business units. With each move, new access had been granted to support the new job. Very little had ever been removed. By the time of our review, the employee could read, and in some cases modify, data in systems belonging to departments they had not worked in for a decade. Every individual grant had been requested, approved, and recorded. Every annual access review had been completed and signed off by a manager. And yet the cumulative result bore almost no relationship to what the employee actually needed to do their job.

I have seen variations of that story in nearly every large organization I have worked with. It illustrates a problem that I believe is among the most widespread and underappreciated in enterprise security: authorization tends to accumulate. Access is easy to grant, difficult to remove, and hard to see in aggregate. Over time, the gap between what identities are permitted to do and what they actually need to do grows wider, and every unnecessary permission becomes an opportunity for an attacker, a careless mistake, or an insider to cause harm.

This chapter examines authorization and access governance. It begins with the distinction between authentication and authorization, compares role-based and attribute-based access control, explores least privilege, separation of duties, and just-in-time access, describes the role of identity governance and administration in managing the lifecycle of access, and closes with practical strategies for reducing excessive permissions and access sprawl.

Authentication Versus Authorization

Authentication and authorization are often discussed together, and the distinction between them can blur in casual conversation. It is worth stating plainly. Authentication establishes who or what is making a request. Authorization determines what that identity is permitted to do. Chapter 4 focused on making the first question trustworthy. This chapter focuses on making the answer to the second question appropriate.

The two are related in an important way. Strong authentication reduces the likelihood that an attacker can assume an identity, but it does nothing to limit the damage if they succeed, or if the legitimate holder of the identity misuses it. Authorization determines the blast radius of a compromised or misused identity. An attacker who phishes a user

with narrowly scoped access gains a limited foothold. An attacker who phishes a user with broad, accumulated access may gain the ability to reach sensitive data and systems across the enterprise.

In my experience, many organizations have invested more heavily in authentication than in authorization over the past several years, partly because authentication improvements are visible and measurable, and partly because authorization problems are distributed across hundreds or thousands of applications, each with its own permission model. Authorization is harder to fix centrally. It is also, I would argue, where a large share of residual identity risk now sits.

Authorization decisions occur at several levels. Coarse-grained decisions determine whether an identity can access an application at all, and they are often enforced at the identity provider through group membership or conditional access policy. Fine-grained decisions determine what the identity can do within the application: which records it can see, which transactions it can approve, which configurations it can change. Fine-grained authorization is usually implemented inside each application, which is precisely why it is so difficult to govern consistently.

Role-Based and Attribute-Based Access Control

Organizations have developed several models for deciding how permissions are assigned. The two most widely discussed are role-based access control and attribute-based access control.

Role-Based Access Control

Role-based access control (RBAC) groups permissions into roles that correspond to job functions.¹ Users are assigned roles, and through those roles they receive the permissions needed to perform their work. A new accounts payable clerk is assigned the accounts payable clerk role and receives the access associated with it, rather than requesting each permission individually.

RBAC has significant advantages. It is intuitive for business managers, it simplifies provisioning and reviews, and it supports regulatory requirements by making access patterns easier to explain and audit. Most enterprise applications and cloud platforms support some form of it.

-
1. David F. Ferraiolo and D. Richard Kuhn, "Role-Based Access Controls," *Proceedings of the 15th National Computer Security Conference*, National Institute of Standards and Technology and National Computer Security Center, Baltimore, October 13-16, 1992, pp. 554-563, <<https://csrc.nist.gov/pubs/conference/1992/10/13/rolebased-access-controls/final>>, accessed October 22, 2026.

RBAC also has well-known weaknesses. Roles tend to proliferate. Organizations create new roles to handle exceptions, regional variations, and special projects, and over time the number of roles can approach or even exceed the number of users, at which point the model no longer simplifies anything. This phenomenon is often called role explosion. Roles also tend to drift: the permissions within a role expand as people request additions, and few organizations regularly review whether each role still reflects what its members actually need. In several enterprises I have worked with, role engineering projects intended to rationalize access stalled because the organization could not agree on what its job functions actually were.

Attribute-Based Access Control

Attribute-based access control (ABAC) makes authorization decisions by evaluating attributes of the subject, the resource, the action, and the environment against policies. Rather than asking whether a user holds a particular role, the system might ask whether the user's department matches the resource's owning department, whether the user's clearance level meets the data's classification, whether the request originates from a managed device, and whether the time falls

within business hours. NIST Special Publication 800-162 provides a widely referenced description of the model.²

ABAC offers flexibility and precision that RBAC struggles to match. Policies can express nuanced conditions without creating a new role for every combination, and access can adapt automatically as attributes change. When an employee transfers departments and their department attribute is updated, their access can change accordingly without manual intervention.

The difficulty is that ABAC depends heavily on the quality and consistency of attributes. If department codes in the HR system are inaccurate, if resources are not reliably tagged with classification and ownership, or if different systems use different attribute definitions, ABAC policies will make incorrect decisions. ABAC policies can also become complex and difficult for business owners to understand, which complicates review and audit.

Other Models

Two further approaches are worth noting. Relationship-based access control determines access from the relationships between entities, such as whether a user is a member of a project that owns a document,

2. Vincent C. Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone, *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, NIST Special Publication 800-162, National Institute of Standards and Technology, Gaithersburg, Md., January 2014 (updated February 2019), <<https://doi.org/10.6028/NIST.SP.800-162>>, accessed October 22, 2026.

and it underpins the sharing models of many collaboration platforms. Policy-based approaches externalize authorization logic from applications into dedicated policy engines, using policy languages that can be centrally managed, version-controlled, tested, and audited. Open-source and commercial policy engines have made this approach increasingly practical, particularly for cloud-native applications and, as Chapter 9 discusses, for constraining the actions of AI agents.³

ARCHITECTURE DECISIONS AND TRADE-OFFS

RBAC or ABAC? In practice, this is rarely an either-or decision. RBAC offers simplicity, business comprehensibility, and strong support in existing tools, but it struggles with fine-grained or context-dependent decisions and tends toward role explosion. ABAC offers precision and adaptability but depends on high-quality attribute data and can produce policies that are difficult to review.

The approach I have seen work most consistently in large enterprises is a hybrid. Use a relatively small number of well-defined roles to establish baseline, coarse-grained access aligned with job functions.

3. Cloud Native Computing Foundation, "Cloud Native Computing Foundation Announces Open Policy Agent Graduation," press release, February 4, 2021, <<https://www.cncf.io/announcements/2021/02/04/cloud-native-computing-foundation-announces-open-policy-agent-graduation/>>, accessed October 22, 2026; Amazon Web Services, "Introducing Cedar, an Open-Source Language for Access Control," *What's New with AWS*, May 10, 2023, <<https://aws.amazon.com/about-aws/whats-new/2023/05/cedar-open-source-language-access-control/>>, accessed October 22, 2026.

Layer attribute-based conditions on top for sensitive resources and context, such as device state, data classification, location, and time. Externalize fine-grained authorization logic into policy engines for new applications where possible. And invest in the attribute data quality that any of these models ultimately depends on.

Least Privilege, Separation of Duties, and Just-in-Time Access

Least Privilege

The principle of least privilege holds that every identity should have only the access necessary to perform its legitimate function, and no more. Few security principles are more widely endorsed, and few are more consistently violated.

The reasons are understandable. Determining the minimum necessary access requires detailed knowledge of what each identity actually does, which application owners and managers often do not have. Granting broad access is faster and avoids the support calls that follow when a permission turns out to be missing. Removing access feels risky because it might break something. And the costs of excessive access are invisible until something goes wrong, while the costs of insufficient access are felt immediately by frustrated users and their managers.

In cloud environments, the gap between granted and used permissions can be especially large. Cloud platforms offer thousands of granular permissions, and administrators frequently assign broad managed policies rather than crafting narrow custom ones. Analyses of cloud environments, including those published by cloud security vendors, have repeatedly suggested that most identities use only a small fraction

of the permissions they hold.⁴ I would treat any specific percentage with caution, but the pattern is consistent with what I have observed directly.

Achieving least privilege is therefore less a matter of policy than of process and data. It requires visibility into which permissions are actually used, mechanisms to remove unused access safely, and a provisioning model that grants narrow access by default and makes broader access the exception.

Separation of Duties

Separation of duties ensures that no single identity can complete a sensitive process from beginning to end without the involvement of another. The person who creates a vendor in the payment system should not be the person who approves payments to that vendor. The developer who writes code should not be able to deploy it to production without review. The administrator who manages audit logs should not be able to delete records of their own actions.

Separation of duties controls are central to fraud prevention and are often required by financial regulations and audit frameworks. They are also, in my experience, frequently undermined in practice. Toxic combinations of access arise when users accumulate roles over time, when emergency access is granted and never removed, and when the same individual holds both business and administrative access to a system. Detecting these conflicts requires a defined set of separation of duties rules and the ability to evaluate access across multiple systems,

4. Microsoft Corporation, *2023 State of Cloud Permissions Risks Report*, Microsoft, Redmond, Wash., March 2023.

since a conflict may involve permissions held in two different applications.

Just-in-Time Access

Standing access, meaning permissions that persist continuously whether or not they are being used, is one of the most significant contributors to identity risk. Every standing privilege is available to an attacker who compromises the identity at any moment, day or night.

Just-in-time (JIT) access reverses this default. Instead of holding sensitive permissions permanently, an identity requests elevated access when it needs it, receives it for a limited time, often after approval or additional authentication, and loses it automatically when the time expires. Zero standing privilege is the logical extension of this approach, in which no identity holds sensitive access by default.

JIT access has become increasingly practical with the maturation of privileged access management tools and native capabilities in major cloud platforms and identity providers. It is most commonly applied to administrative access, which Chapter 6 discusses in detail, but the same principle can be extended to access to sensitive data, production systems, and high-impact business functions. In my experience, organizations that adopt JIT access for even a subset of their most sensitive permissions often see a substantial reduction in their exposure with a relatively modest operational burden, provided the request and approval process is fast enough that users do not seek workarounds.

ENTERPRISE FIELD NOTES

When introducing just-in-time access, the most common objection I hear is that it will slow people down. In practice, the friction tends to be concentrated in the first few weeks, while

workflows and approval paths are adjusted. Once the process is fast and predictable, most users adapt quickly, and many come to prefer it because it removes the anxiety of holding powerful access they rarely use. The programs that struggled were usually those that required manual approvals from busy managers for routine, low-risk elevations. Reserving human approval for genuinely sensitive access, and allowing self-service elevation with strong authentication and logging for routine tasks, tended to resolve that problem.

Identity Governance and Administration

Identity governance and administration (IGA) encompasses the processes and tools that manage identities and their access throughout their lifecycle. Its core functions include provisioning and deprovisioning, access requests and approvals, access certification or review, role and policy management, separation of duties enforcement, and reporting for audit and compliance.

The Identity Lifecycle

The identity lifecycle is often described in terms of joiners, movers, and leavers. Joiners need access provisioned quickly and accurately when they start. Movers, people who change roles, departments, or locations, need new access granted and old access removed. Leavers need all access revoked promptly when they depart.

Of these three events, my experience suggests that movers are handled worst. Joiner processes are usually well understood because a new employee who cannot work generates immediate complaints.

Leaver processes receive attention because auditors check them. Mover processes, however, frequently add new access without removing old access, which is exactly how the accumulation described at the start of this chapter occurs. Automating the removal of access tied to a previous role, often after a short transition period, is one of the most effective improvements an IGA program can make.

Contractors, partners, and other non-employees present additional difficulties. They are often managed outside the HR system, their engagements may end without any formal notification to IT, and their access may persist long after their work is complete. Establishing an authoritative source for non-employee identities, with defined sponsors and end dates, is an essential and frequently neglected element of governance.

Access Reviews

Periodic access reviews, in which managers or application owners certify that users' access remains appropriate, are a staple of identity governance and a common regulatory requirement. They are also, in many organizations, a source of considerable frustration and limited security value.

The typical failure mode is easy to recognize. A manager receives a long list of entitlements described in cryptic technical names, with little context about what each one permits. They have dozens of employees to review and a deadline. Revoking access risks disrupting someone's work. Approving everything takes a few clicks. The result, sometimes called rubber-stamping, is a review process that satisfies auditors while doing little to reduce actual risk.

Improving reviews tends to involve several steps: presenting entitlements in business-friendly language; highlighting the access that

matters most, such as privileged, sensitive, or unused permissions, rather than asking reviewers to evaluate everything equally; providing usage data so reviewers can see whether access has been exercised recently; focusing reviews on changes and anomalies rather than full recertification of unchanged access; and measuring revocation rates and review quality rather than simply completion rates. Chapter 11 discusses how AI can assist with some of these tasks, along with the risks of relying on it uncritically.

Connectivity and Coverage

As Chapter 3 noted, an IGA platform can only govern what it is connected to. In many enterprises, the governance program covers the major directories and a set of core applications, while hundreds of SaaS applications, departmental systems, and cloud environments sit outside its reach. Access in those systems is managed manually, if at all. Expanding coverage, prioritized by the sensitivity of each application, is often the single largest determinant of whether governance delivers real security value.

BEST PRACTICES THAT WORK

Measure your governance program by outcomes rather than activity. Completion rates for access reviews and the number of applications onboarded are useful operational metrics, but they say little about risk. More meaningful indicators include the time between an employee's termination and the removal of their access across all systems; the number of active accounts belonging to departed employees or expired contractors; the proportion of privileged access that is standing rather than just-in-time; the number of unresolved separation of duties conflicts; and the vol-

ume of granted permissions that have not been used in the past ninety days. These measures tend to reveal whether governance is reducing exposure or merely documenting it.

Reducing Excessive Permissions, Access Sprawl, and Entitlement Risk

Most organizations are not starting from a clean slate. They have years of accumulated access to address, and they cannot pause the business while they redesign their authorization model. Reducing existing entitlement risk is therefore a practical, incremental effort.

The first step is visibility. Organizations need to understand what access exists, who holds it, and, wherever possible, whether it is used. Usage data is particularly valuable because it allows the organization to distinguish between access that is needed and access that merely exists. Many cloud platforms provide information about when permissions were last used, and cloud infrastructure entitlement management tools can analyze effective permissions across complex cloud policy structures. For on-premises applications, usage data may be harder to obtain and may require analysis of application logs.

The second step is prioritization. Not all excessive access carries equal risk. Unused administrative privileges, access to sensitive data, permissions that enable privilege escalation, access held by departed users or dormant accounts, and separation of duties conflicts in financial processes generally deserve attention before broad read access to low-sensitivity systems.

The third step is remediation, and here caution pays off. Removing access that turns out to be needed can disrupt business operations and erode trust in the security program. Approaches that reduce this risk

include removing access first from clearly dormant accounts; disabling rather than deleting access initially so it can be quickly restored; notifying users and owners in advance; removing access in stages, starting with permissions that have not been used for an extended period; and establishing a fast path for restoring access that turns out to be needed.

The fourth step is prevention. Cleaning up existing access is of limited value if new access continues to accumulate in the same way. Durable improvement requires changing the provisioning model so that access is narrower by default, time-limited where appropriate, tied to roles or attributes that change automatically when a person's situation changes, and subject to automatic expiry when unused.

Finally, I would emphasize that non-human identities must be included in this work. Service accounts, application identities, and integrations often hold some of the broadest and least reviewed permissions in the environment. Governance programs that focus exclusively on human users may address only part of the problem. Chapter 8 discusses non-human identities in depth.

REALISTIC THREAT SCENARIO

A marketing analyst at a retail company falls for a well-crafted phishing message, and the attacker obtains a valid session. The analyst's current role requires access only to campaign reporting tools. However, the analyst previously worked in customer service and, before that, on a data integration project. Access from both roles was never removed.

Using the analyst's session, the attacker discovers that the account can still query the customer support platform, including customer contact details and order histories, and can read files in a shared data repository used by the old integration project, where an export of customer records and a configuration file

containing an API key had been left behind. The API key grants read access to the company's loyalty program database.

Within a few hours, the attacker exfiltrates a large volume of customer data. The initial compromise affected an identity with seemingly modest privileges. The damage resulted from accumulated access and stored secrets that had nothing to do with the analyst's current job.

Automated removal of access on role change, periodic review of unused entitlements, scanning of shared repositories for credentials and sensitive exports, and anomaly detection on unusual data queries from an identity whose role did not require them would each have reduced the impact substantially.

PRACTITIONER CHECKLIST

Can you identify, for any given user, all of the access they hold across directories, cloud platforms, and major applications?

When employees change roles, is access from their previous role removed automatically, and within what time frame?

Is there an authoritative source for contractors and other non-employees, with sponsors and end dates?

How many roles exist in your environment relative to the number of users, and when were role definitions last reviewed?

Do access reviews present usage data and focus reviewers on high-risk access, and what proportion of reviewed access is actually revoked?

Are separation of duties rules defined and evaluated across systems, particularly for financial processes?

What proportion of sensitive and administrative access is standing rather than just-in-time?

Do you have visibility into unused permissions in cloud environments, and a safe process for removing them?

Are service accounts and other non-human identities included in governance processes?

Looking Ahead

Authorization controls determine how much damage a compromised identity can do. Nowhere is that question more consequential than for privileged identities: the administrators, service accounts, and secrets that can control infrastructure, change security configurations, and access the organization's most sensitive systems.

Chapter 6 focuses on these high-value identities. It examines how to protect administrator accounts, how privileged access management platforms are architected and deployed, how to manage service accounts, API keys, certificates, and application secrets, and how vaulting, credential rotation, ephemeral credentials, and just-in-time privilege can limit the impact when, despite our best efforts, an identity is compromised.

6. Privileged Access and Secrets Management

During an incident response engagement some years ago, I sat with a team trying to answer what should have been a simple question: which credentials did the attacker have? We knew the attacker had obtained domain administrator access. What we did not know, and what took days to establish, was everything else that access had exposed. Administrators had stored passwords for network devices in a spreadsheet on a file share. Scripts on several servers contained embedded credentials for databases and cloud accounts. A deployment tool held API keys for the production environment. A service account used by the backup system had rights to nearly every server in the company, and its password had not been changed since the system was installed. Each of these had to be identified, assessed, and rotated before we could be confident the attacker had been

removed. Some of the rotations broke production systems because nobody knew what depended on them.

That engagement reinforced a conviction I have held ever since: privileged access is where identity risk is most concentrated, and secrets are where it is most likely to hide. An ordinary user account, if compromised, may expose a mailbox and a handful of applications. A privileged account or an exposed secret may expose the entire enterprise. This chapter examines how to protect administrator accounts and other high-value identities, how privileged access management platforms are designed and deployed, how to manage the service accounts, API keys, certificates, and application secrets that pervade modern environments, and how vaulting, rotation, ephemeral credentials, and just-in-time privilege can limit the damage when an identity is compromised.

Protecting Administrator Accounts and High-Value Identities

What Counts as Privileged

The traditional definition of privileged access focuses on administrators: domain administrators, root accounts on servers, database administrators, network device administrators, and the administrators of cloud platforms and the identity provider itself. These accounts can change configurations, create and modify other identities, access or alter data directly, and disable security controls. Their compromise is generally the turning point in a serious intrusion.

In my experience, a narrow definition of privilege leaves significant gaps. A broader view of high-value identities includes several addi-

tional categories. Business users who can approve payments, change supplier banking details, or access large volumes of sensitive data may represent greater financial risk than many system administrators. Help desk staff who can reset passwords and MFA registrations hold, in effect, the ability to take over other accounts. Developers and DevOps engineers with access to source code, build pipelines, and production deployment can introduce malicious changes that affect every customer. Executives and their assistants are targeted for impersonation and fraud. Security team members themselves often hold broad visibility and administrative access to security tools that attackers would very much like to disable.

I would encourage security teams to build an explicit inventory of high-value identities that spans all of these categories, rather than relying on membership in a handful of administrative groups. As Chapter 2 noted, effective privilege frequently exceeds what group names suggest, particularly when nested groups, delegated permissions, and cloud IAM policies are taken into account.

Core Protections for Administrative Accounts

Several practices form the foundation of administrator account security, and I find that they are still inconsistently applied even in mature organizations.

Administrators should use separate accounts for privileged and ordinary work. Reading email and browsing the web with an account that holds administrative rights exposes those rights to phishing and malicious content. A dedicated administrative account, used only for administrative tasks, substantially reduces that exposure.

Privileged accounts should require phishing-resistant authentication without exception. As Chapter 4 discussed, these are the accounts

for which the investment in hardware security keys or device-bound passkeys is most clearly justified.

Administrative work, particularly for tier-zero systems such as the identity provider and directory, should be performed from privileged access workstations: hardened devices dedicated to administration, without email or general web browsing, and with tightly controlled software.¹ The reason is straightforward. If an administrator signs in to a sensitive system from a compromised laptop, the attacker on that laptop can potentially capture the credentials or session. Protecting the device used for administration is as important as protecting the credential itself.

The tiered administration model introduced in Chapter 3 extends this logic. Credentials for the highest tier should never be exposed on lower-tier systems where they might be harvested. Administrators of domain controllers should not sign in to ordinary servers or workstations with those credentials, and workstation administrators should not hold rights over servers. Maintaining this separation requires discipline and periodic verification, because it tends to erode under operational pressure.

Finally, the number of privileged accounts should be minimized. Every standing administrative account is a target. Reducing their number, combined with the just-in-time approaches discussed later in this chapter, is one of the most effective ways to shrink the attack surface.

1. Microsoft Corporation, "Privileged Access Devices," *Microsoft Learn*, <<https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-devices>>, accessed October 22, 2026.

ENTERPRISE FIELD NOTES

When I ask administrators why they continue to use privileged accounts for everyday work, the answer is almost always convenience rather than ignorance. Switching between accounts, or between an ordinary laptop and a privileged access workstation, adds friction to tasks they perform many times a day. Programs that succeeded in changing this behavior tended to reduce that friction deliberately: providing virtualized privileged workstations that could be launched quickly, streamlining just-in-time elevation, and involving senior administrators in designing the workflow. Programs that simply issued a policy and expected compliance generally found that exceptions accumulated within months.

Privileged Access Management: Architecture and Implementation

Privileged access management (PAM) platforms provide centralized control over privileged credentials and sessions. Although products differ, most PAM architectures include several common capabilities.

A credential vault stores privileged passwords, keys, and other secrets in encrypted form, removing them from spreadsheets, scripts, and administrators' memories. Access to vaulted credentials is controlled by policy and logged.

Credential management automates the rotation of privileged passwords, often after each use or on a defined schedule, so that a credential obtained by an attacker quickly becomes invalid. Some platforms

check out credentials to users for a limited period and rotate them on check-in.

Session management brokers privileged connections to target systems. Rather than receiving the actual credential, the administrator connects through the PAM platform, which injects the credential into the session on their behalf. The administrator never sees the password, which means it cannot be phished, written down, or reused elsewhere. Sessions can be monitored in real time and recorded for later review.

Elevation and delegation capabilities allow users to run specific commands or applications with elevated rights on endpoints and servers without holding full administrative access, which is particularly useful for removing local administrator rights from workstations.

Just-in-time access workflows grant privileged rights for a limited time, often after approval, and revoke them automatically. Many PAM platforms, identity providers, and cloud platforms now offer this capability natively.

Implementation Realities

PAM implementations have a mixed reputation, and in my experience the difficulties are rarely technical. The most common problem is incomplete coverage. Organizations purchase a PAM platform, onboard a set of domain administrator accounts and critical servers, and declare the initial phase complete. Years later, a large share of privileged accounts, cloud administrative roles, network devices, databases, and SaaS administrator accounts remain outside the platform. Attackers do not limit themselves to the onboarded systems.

A second common problem is that PAM becomes a vault for credentials that are still standing and still broadly privileged. Vaulting a domain administrator password is an improvement over storing it in

a spreadsheet, but if dozens of administrators can check it out at any time without approval, the underlying exposure has changed less than it appears.

A third problem is operational friction. If connecting through the PAM platform is slow or unreliable, administrators find ways around it, often by creating local accounts or caching credentials. PAM platforms themselves must also be treated as tier-zero infrastructure. A compromised PAM system may expose every credential it protects, so its own administration, availability, and recovery deserve careful design.

The implementations I have seen succeed tended to share a few characteristics: they began with a clear inventory of privileged accounts and prioritized them by risk; they measured coverage honestly; they combined vaulting with just-in-time access rather than treating vaulting as the end state; they extended PAM to cloud and SaaS administration rather than focusing only on on-premises systems; and they invested in administrator experience so that the secure path was also the convenient one.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Session brokering or credential checkout? Session brokering, in which the PAM platform injects credentials and administrators never see them, provides strong protection against credential theft and enables recording and real-time monitoring. It can, however, introduce latency, compatibility problems with some tools and protocols, and a dependency on the PAM platform's availability.

Credential checkout, in which administrators retrieve a credential and use it directly, is more flexible and compatible, but

the credential is exposed to the administrator's device and memory during use.

A common and, in my view, sensible approach is to use session brokering for tier-zero systems and other highly sensitive targets, where the security benefit justifies the operational overhead, and to use checkout with automatic rotation after use for lower-tier systems or tools that cannot be brokered. Wherever possible, the long-term direction should be toward ephemeral, just-in-time credentials that reduce the need for either.

Service Accounts, API Keys, Certificates, and Application Secrets

Human administrators are only part of the privileged access picture. Applications, scripts, and infrastructure also need credentials to authenticate to one another, and these machine credentials are frequently among the most powerful and least protected in the environment. Chapter 8 addresses non-human identities as a governance challenge. Here, I focus on the credentials themselves and how they are managed.

Service Accounts

Service accounts are identities used by applications and services rather than by people. In on-premises environments, they are often directory accounts with passwords that applications use to connect to databases, file shares, and other services. Their weaknesses are well known: broad permissions granted for convenience, passwords that are never changed because nobody knows what will break, interactive sign-in rights that allow attackers to use them like ordinary accounts, and

a lack of clear ownership. Directory platforms provide mechanisms, such as group managed service accounts in Active Directory, that handle password rotation automatically, and I would encourage organizations to adopt them wherever applications support them.²

API Keys and Tokens

API keys and tokens authenticate applications to cloud services, SaaS platforms, and internal APIs. Many are long-lived, highly privileged, and stored in places they should not be: source code repositories, configuration files, container images, CI/CD variables, chat messages, and documentation pages. Attackers actively scan public repositories for exposed keys, and in some reported cases exposed cloud keys have been abused within minutes of publication.³ Secrets that are committed to a repository can persist in its history long after they are removed from the current code.

2. Microsoft Corporation, "Group Managed Service Accounts Overview," *Microsoft Learn*, <<https://learn.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview>>, accessed October 22, 2026.

3. William Gamazo and Nathaniel Quist, "CloudKeys in the Air: Tracking Malicious Operations of Exposed IAM Keys," *Unit 42*, Palo Alto Networks, October 30, 2023, <<https://unit42.paloaltonetworks.com/malicious-operations-of-exposed-iam-keys-cryptojacking/>>, accessed October 22, 2026.

Certificates

Digital certificates authenticate servers, devices, and in some cases users and workloads. They are often managed separately from other identity functions, sometimes by network or infrastructure teams with their own tools and spreadsheets. Two problems recur. The first is unexpected expiry, which has caused significant outages at many organizations when a forgotten certificate lapsed. The second is compromise or misuse of private keys, which can allow attackers to impersonate trusted services or sign malicious code. The industry is moving toward substantially shorter certificate lifetimes for public TLS certificates, a change that is likely to make automated certificate lifecycle management effectively mandatory for most enterprises.⁴

Secrets Sprawl

The cumulative result of all these credentials is what is often called secrets sprawl: secrets distributed across code, configuration, infrastructure, and collaboration tools, with no complete inventory, inconsistent protection, and little visibility into who or what can access them. In my experience, organizations that conduct their first

4. CA/Browser Forum, "Ballot SC-081v3: Introduce Schedule of Reducing Validity and Data Reuse Periods," CA/Browser Forum, April 11, 2025, <<https://cabforum.org/2025/04/11/ballot-sc081v3-introduce-schedule-of-reducing-validity-and-data-reuse-periods/>>, accessed October 22, 2026.

comprehensive secrets scan are frequently surprised by the volume of exposed credentials they find, including credentials for production systems and cloud accounts with administrative rights.⁵

BEST PRACTICES THAT WORK

Treat secret scanning as a continuous control rather than a one-time cleanup. In practice, this tends to mean scanning code at the point of commit so that secrets are caught before they enter a repository; scanning existing repositories, including their history; scanning container images, infrastructure-as-code templates, CI/CD configurations, and collaboration platforms; and, critically, treating any discovered secret as compromised and rotating it, rather than simply deleting it from the file where it was found. Removing a secret from the current version of a file does not remove it from history, backups, or the hands of anyone who has already copied it.

Vaulting, Rotation, Ephemeral Credentials, and Just-in-Time Privilege

The tools for managing secrets have evolved through several stages, and I find it useful to think of them as a progression, each step reducing exposure further than the last.

5. GitGuardian, *The State of Secrets Sprawl 2025*, GitGuardian, March 2025, <<https://blog.gitguardian.com/the-state-of-secrets-sprawl-2025/>>, accessed October 22, 2026.

Vaulting

The first step is to move secrets out of code, configuration files, and documents into a dedicated secrets manager or vault. Applications retrieve secrets at runtime through an authenticated API, and access to each secret is controlled by policy and logged. Most major cloud platforms offer native secrets management services, and independent secrets management platforms support hybrid and multi-cloud environments. Vaulting provides central visibility and control, and it removes the most obvious exposure. It does not by itself address the fact that the secrets may still be long-lived and highly privileged.

Rotation

The second step is to rotate secrets regularly and automatically, so that a stolen secret has a limited useful lifetime. Rotation is conceptually simple and operationally challenging, because every consumer of a secret must be able to obtain the new value without disruption. In my experience, the fear of breaking production is the most common reason rotation is not implemented, and it is a reasonable fear when dependencies are poorly understood. Mapping which applications use each secret, and building applications to retrieve secrets dynamically rather than caching them indefinitely, are prerequisites for rotation at scale.

Ephemeral and Dynamic Credentials

The third step eliminates long-lived secrets altogether wherever possible. Dynamic credentials are generated on demand for a specific consumer, with a short lifetime, and are revoked automatically when

they expire. A secrets platform might create a database credential for an application at startup that expires after an hour, for example. Cloud platforms offer short-lived tokens obtained by assuming roles rather than using static access keys. Workload identity mechanisms, which Chapter 8 discusses in more detail, allow workloads to authenticate based on their verified runtime identity and to exchange that identity for short-lived credentials, without any stored secret at all. Workload identity federation allows CI/CD pipelines and workloads running in one environment to obtain short-lived credentials in another based on trust between identity providers, rather than through shared keys.

In my view, ephemeral credentials represent the most significant improvement available for machine credential security. A credential that exists for minutes and is scoped to a single purpose offers far less value to an attacker than one that persists for years with broad permissions.

Just-in-Time Privilege

For human administrators, the equivalent of ephemeral credentials is just-in-time privilege, introduced in Chapter 5. Administrators hold no standing privileged rights. When they need to perform a privileged task, they request elevation, authenticate strongly, receive approval where required, and are granted the necessary rights for a limited period. Major cloud platforms and identity providers increasingly support this model natively for their own administrative roles. Combined with privileged access workstations and session monitoring, just-in-time privilege can bring an organization close to the goal of zero standing privilege for its most sensitive access.

Preventing Privilege Escalation and Limiting the Impact of Compromise

Despite the best protections, some identities will be compromised. The final objective of privileged access and secrets management is to make it difficult for an attacker to turn a limited compromise into a broad one, and to limit the damage if they succeed.

Several measures work together toward that objective. Removing standing privilege means that compromising an ordinary account does not automatically yield administrative rights. Eliminating stored secrets from file shares, scripts, and repositories closes some of the most common escalation paths. Enforcing tiering prevents high-tier credentials from being exposed on systems where attackers are likely to land first. Analyzing attack paths, using tools that map the relationships between identities, groups, permissions, and systems, allows defenders to identify and break escalation chains before attackers find them. Restricting service account permissions and preventing interactive sign-in for service accounts limits their usefulness to an attacker.

Monitoring is equally important. Privileged activity should generate high-fidelity telemetry: elevation requests, credential checkouts, session recordings, changes to privileged group memberships, creation of new privileged accounts or keys, and changes to PAM and vault configurations. Unusual patterns, such as privileged access outside normal hours, use of break-glass accounts, or retrieval of large numbers of secrets in a short period, deserve prompt investigation. Chapter 7 discusses identity threat detection and response in depth.

Finally, organizations should plan for the scenario in which privileged credentials are compromised at scale. This means knowing which secrets exist, which systems depend on them, and how to rotate them quickly. In the incident I described at the start of this chapter, the

slowest part of recovery was not removing the attacker's access; it was discovering and safely rotating every credential that might have been exposed. Organizations that have inventoried their secrets, automated rotation, and practiced mass credential resets can recover far faster than those attempting it for the first time under pressure.

REALISTIC THREAT SCENARIO

A software company's developer accidentally commits a configuration file containing a cloud access key to a public repository. The key belongs to a service identity used by a deployment pipeline and holds broad permissions in the production cloud account, granted during an earlier migration and never reduced. Automated scanners operated by attackers detect the key shortly after it is published.

The attacker uses the key to enumerate the cloud environment, discovers a secrets manager, and finds that the pipeline identity is permitted to read every secret in it, including database credentials and API keys for third-party payment and messaging services. The attacker also creates a new access key for a separate administrative identity, establishing persistence that will survive revocation of the original key.

The company's secret scanning tool alerts on the exposed key, and the security team revokes it within a few hours. However, because the team treats the incident as a single leaked key, it does not immediately investigate what the key was used for, and the newly created administrative key goes unnoticed for several days.

Using workload identity federation for the pipeline instead of a static key, scoping the pipeline's permissions to deployment tasks only, restricting secret access to the specific secrets each identity requires, alerting on creation of new access keys for ad-

ministrative identities, and treating any exposed privileged credential as a full incident requiring investigation of its use would each have reduced the impact considerably.

PRACTITIONER CHECKLIST

Do you maintain an inventory of high-value identities that includes business approvers, help desk staff, developers, executives, and security personnel as well as system administrators?

Do administrators use separate accounts and privileged access workstations for administrative tasks, particularly for tier-zero systems?

What proportion of privileged accounts, including cloud, SaaS, network, and database administration, is covered by your PAM platform?

How much privileged access is standing, and how much is granted just in time?

Is your PAM platform itself protected, monitored, and recoverable as tier-zero infrastructure?

Do you scan code, repositories, pipelines, images, and collaboration tools for exposed secrets continuously, and are discovered secrets rotated rather than simply deleted?

Where could static credentials be replaced with ephemeral credentials or workload identity federation?

Are certificates inventoried and renewed automatically, and are you prepared for shorter certificate lifetimes?

Could you identify and rotate every privileged credential in your environment within days if required?

Looking Ahead

The controls described in Part II so far are primarily preventive. They make it harder for attackers to obtain identities and limit what those identities can do. Prevention, however, is never complete. Some credentials will be phished, some tokens will be stolen, and some insiders will misuse their access. When that happens, the organization's ability to detect the compromise quickly and respond effectively becomes decisive.

Chapter 7 addresses that capability. It examines identity threat detection and response, how to establish visibility across authentication, authorization, and identity changes, how to detect anomalous behavior and suspicious privilege use, how to investigate identity incidents and coordinate across teams, and how to contain, recover, and improve after an identity-based attack.

7. Identity Threat Detection, Response, and Recovery

Some years ago, I was called into an organization on a Friday evening after its security operations center noticed something odd: an employee in the finance department had signed in successfully from two countries within the same hour. The analyst on duty did what the playbook said. She reset the user's password and closed the ticket. On Monday, the same account was again active from an unfamiliar location.

When we investigated properly, we found that the attacker had never needed the password after the initial compromise. They held a stolen session token that remained valid after the reset, had registered a second MFA method on the account, had created a mailbox rule to hide security notifications, and had granted consent to a third-party application that could read the user's email indefinitely. Resetting the password had addressed none of these. The playbook had been written

for a world in which a password was the identity, and that world had quietly ended.

That incident illustrates why identity threat detection and response has emerged as a distinct discipline. Traditional security operations grew up around malware, network intrusions, and endpoint activity. Identity-based attacks often involve none of those things. The attacker signs in, uses legitimate tools, and abuses legitimate permissions. Detecting them requires identity telemetry and identity expertise, and responding to them requires actions, such as revoking tokens, removing rogue authentication methods, and repairing directory trust, that many incident response teams were not originally trained to perform.

This chapter examines identity threat detection and response as a set of capabilities and use cases. It discusses how to establish visibility across authentication, authorization, and identity changes; how to detect anomalous behavior, compromised accounts, and suspicious privilege use; how to investigate identity incidents and coordinate response across teams; and how to contain, recover from, and learn from identity-based attacks.

Identity Threat Detection and Response: Capabilities and Use Cases

The term identity threat detection and response, or ITDR, gained prominence when industry analysts began using it around 2022 to describe tools and practices focused specifically on protecting identity

infrastructure and detecting the misuse of identities.¹ Since then, a range of vendors has adopted the label, and the capabilities it describes now appear in identity providers, extended detection and response platforms, dedicated ITDR products, and security information and event management systems. As with Zero Trust, I find it more useful to think of ITDR as a set of capabilities than as a product category.

In my view, a mature ITDR capability addresses four broad objectives.

The first is protecting identity infrastructure itself. Directories, identity providers, federation services, synchronization servers, and PAM platforms are the systems attackers most want to control. ITDR includes continuous assessment of their configuration and posture, such as detecting risky directory settings, dangerous delegated permissions, weak federation configurations, and stale privileged accounts, as well as monitoring for attacks against them.

The second is detecting compromised identities. This includes recognizing signs that credentials, sessions, or tokens are being used by someone other than their legitimate holder, whether a human user, a service account, or an AI agent.

The third is detecting misuse of legitimate access. Some of the most damaging identity incidents involve identities that are not compromised in the usual sense but are being used inappropriately, by

1. Gartner, Inc., "Gartner Identifies Top Security and Risk Management Trends for 2022," press release, March 7, 2022, <<https://www.gartner.com/en/newsroom/press-releases/2022-03-07-gartner-identifies-top-security-and-risk-management-trends-for-2022>>, accessed October 22, 2026.

malicious insiders, by careless users, or by automated systems behaving unexpectedly.

The fourth is enabling effective response. Detection has limited value if the organization cannot act quickly to revoke sessions, disable accounts, remove persistence, and restore trusted identity infrastructure.

Common ITDR use cases include detecting credential stuffing and password spraying; identifying adversary-in-the-middle phishing and session token replay; flagging MFA fatigue attacks and suspicious MFA method registrations; detecting suspicious OAuth application consents; recognizing privilege escalation through group membership or role changes; identifying attacks against the on-premises directory, such as attempts to harvest credential material or forge authentication tickets; detecting changes to federation trusts and identity provider configuration; identifying anomalous service account behavior; and spotting reactivation of dormant accounts. Increasingly, the list also includes detecting unusual behavior by AI agents and the integrations that serve them, a topic Chapter 9 explores in greater depth.

ENTERPRISE FIELD NOTES

When I assess identity detection capabilities, I often ask the security operations team a simple question: if an attacker added a new federation trust to your cloud tenant, or granted a new identity the right to replicate directory data, how long would it take you to know? Many teams can answer confidently for malware on a laptop. Far fewer can answer for these identity infrastructure changes, which may represent a more severe compromise. The gap usually reflects history rather than negligence. Security operations centers were built around endpoint and net-

work telemetry, and identity telemetry was added later, often incompletely.

Establishing Visibility Across Authentication, Authorization, and Identity Changes

Detection depends on data. Before an organization can recognize identity threats, it needs to collect, retain, and correlate the right telemetry. In my experience, gaps in this foundation are the most common reason identity attacks go undetected.

Authentication Telemetry

Authentication logs record who signed in, when, from where, using what device and method, to which application, and with what result. The identity provider is usually the richest source, but it is not the only one. On-premises directory authentication, including Kerberos and NTLM activity, VPN and remote access gateways, privileged access platforms, cloud platform consoles and APIs, and SaaS applications that authenticate locally all generate authentication events. Token issuance and refresh events are particularly important, because, as the incident at the start of this chapter showed, a session may continue long after the initial sign-in.

Authorization and Activity Telemetry

Knowing that an identity signed in is only part of the story. Security teams also need to know what it did: which resources it accessed, which data it read or downloaded, which administrative actions it

performed, and which API calls it made. Cloud platform audit logs, SaaS application audit logs, database activity monitoring, file access logs, and PAM session records provide this information. Coverage is often uneven. Some SaaS applications offer detailed audit logs only at premium license tiers, and some legacy applications log very little.

Identity Change Telemetry

Changes to identities and their permissions are among the most important signals an organization can monitor. These include account creation, deletion, enablement, and disablement; group and role membership changes; registration or removal of MFA methods; password resets and recovery actions; creation of new access keys, secrets, or certificates for identities; OAuth application consents and service principal changes; changes to conditional access and authentication policies; and changes to federation, synchronization, and directory configuration. Many of the persistence techniques described in Chapter 2 appear in this telemetry and nowhere else.

Context and Enrichment

Raw events become far more useful when enriched with context. Which identities are privileged? Who owns each service account? What is the user's role, department, manager, and normal working location? Is the device managed and compliant? Is the source IP address associated with an anonymizing service or known malicious infrastructure? Identity governance data, asset inventories, device management systems, and threat intelligence all contribute to this context. Without it, analysts struggle to distinguish a routine administrative change from an attacker's preparation.

BEST PRACTICES THAT WORK

Build an identity telemetry inventory alongside your application identity inventory. For each identity source, record which events are collected, where they are sent, how long they are retained, and whether they are actively used in detection rules. Pay particular attention to retention. Identity-based intrusions can remain undetected for weeks or months, and investigators frequently need to trace activity back to the initial compromise. Retention periods that are adequate for alerting may be far too short for investigation. Confirm also that the logs of the identity infrastructure are protected against tampering by the same administrators whose actions they record.

Detecting Anomalous Behavior, Compromised Accounts, and Suspicious Privilege Use

With visibility established, detection can take several complementary forms.

Signature and Rule-Based Detection

Some identity attacks leave recognizable patterns. A burst of failed sign-ins across many accounts from a small number of sources suggests password spraying. A large number of push notifications sent to a single user in a short period suggests MFA fatigue. A request for directory replication data from a system that is not a domain controller may indicate an attempt to harvest credentials. A new federation trust,

a new highly privileged role assignment, or a new credential added to an application with broad permissions warrants immediate review. Rule-based detections for well-understood techniques are reliable and explainable, and they form the backbone of most identity detection programs. Frameworks such as MITRE ATT&CK can help organizations map their detection coverage against known techniques.²

Behavioral and Anomaly Detection

Many identity attacks are designed to look normal, so rules alone are insufficient. Behavioral analytics establish baselines for how identities typically behave, including when and where they sign in, which applications they use, how much data they access, and which administrative actions they perform, and then flag significant deviations. Examples include a user accessing applications they have never used, downloading unusually large volumes of data, or signing in from a new country; a service account signing in interactively or from an unexpected host; an administrator performing actions outside their usual scope; and a dormant account suddenly becoming active.

Behavioral detection is powerful but imperfect. Baselines can be noisy in organizations with frequent travel, reorganizations, or seasonal patterns. Attackers who move slowly and operate within the victim's normal patterns may avoid triggering anomalies. False positives can overwhelm analysts and erode confidence in the alerts. In my experience, behavioral detections work best when they are combined with contextual risk factors, such as whether the identity is privileged

2. The MITRE Corporation, *MITRE ATT&CK*, <<https://attack.mitre.org/>>, accessed October 22, 2026.

or the accessed data is sensitive, rather than treated as standalone signals. Chapter 11 discusses how machine learning can improve these capabilities and the limitations that come with it.

Posture-Based Detection

A third form of detection identifies weaknesses before they are exploited. Continuous assessment of identity posture can surface accounts without MFA, privileged accounts with stale passwords, service accounts with interactive sign-in rights, risky delegations in the directory, overly permissive cloud roles, applications with excessive API permissions, and attack paths that would allow an ordinary account to reach tier-zero privileges. These are not incidents in themselves, but they show where incidents are likely to occur, and they help security teams prioritize both prevention and monitoring.

Detecting Suspicious Privilege Use

Privileged activity deserves its own detection focus because its consequences are so severe. Useful signals include elevation requests outside normal working patterns; privileged sessions from unexpected devices or locations; use of break-glass accounts, which should be rare and should trigger an immediate alert; bulk changes to permissions or group memberships; disabling of logging, security tools, or policies; creation of new privileged identities; and retrieval of an unusual number of secrets from a vault. Because privileged actions are relatively infrequent in a well-designed environment, anomalies are often easier to spot than they are for ordinary user activity.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Native identity provider detection or a centralized detection platform? Identity providers and cloud platforms increasingly offer built-in risk detection, which benefits from deep visibility into their own signals and from intelligence drawn across many customers. These capabilities are often quick to deploy and well tuned.

They also have limitations. Each provider sees only its own domain, so an attack that moves from an on-premises directory to a cloud tenant to a SaaS application may appear as unrelated fragments. Correlating identity signals with endpoint, network, and application telemetry generally requires a centralized platform such as a SIEM or extended detection and response system.

The approach I have seen work best uses native detections as high-quality signals and feeds them, together with raw identity telemetry, into a central platform where cross-domain correlation, enrichment, and response orchestration occur. Organizations should be careful not to assume that enabling a vendor's risk detection constitutes a complete identity detection program.

Investigating Identity Incidents and Coordinating Response

Identity incidents present distinctive investigative challenges. The attacker's activity often appears legitimate in isolation. Evidence is distributed across many systems, each with its own log format, retention period, and time conventions. And the scope of a compromise can be difficult to establish, because a single compromised identity may

have touched dozens of applications, and a compromised privileged identity may have created further footholds.

Key Investigative Questions

In my experience, identity investigations tend to revolve around a consistent set of questions. How was the identity first compromised: phishing, token theft, credential reuse, help desk manipulation, or an exposed secret? When did the compromise begin? What did the attacker do with the identity, including which applications they accessed, which data they viewed or exfiltrated, and which actions they took? Did they obtain additional identities or privileges? What persistence did they establish, such as new accounts, new MFA methods, new OAuth consents, new access keys, new mail rules, or changes to federation or directory configuration? And is the attacker still present?

Answering these questions requires the ability to construct a timeline of an identity's activity across systems, which depends on the telemetry and retention discussed earlier. It also requires analysts who understand identity systems well enough to recognize what is normal and what is not. In many organizations, that expertise sits in the IAM team rather than the security operations center.

Coordinating Across Teams

Identity incidents almost always span organizational boundaries. The security operations center detects and investigates. The IAM team understands the directories, identity providers, and governance tools, and often holds the administrative rights needed to take response actions. Infrastructure and cloud teams manage the systems that identities accessed. Application owners understand what activity is normal

within their applications. The help desk may be involved in the initial compromise and will certainly be involved in restoring access for affected users. Legal, privacy, communications, and business leadership become involved as the incident's impact becomes clear.

I have seen identity incidents slowed considerably by unclear responsibilities, particularly between security operations and IAM teams. Who has the authority to disable an executive's account in the middle of the day? Who can revoke all sessions for a business-critical service account? Who decides whether to rotate a federation signing certificate, an action that may disrupt access for the entire organization? These questions are best answered in advance, through documented playbooks, defined decision rights, and joint exercises.

One further consideration is often overlooked. If an attacker has compromised identity infrastructure or privileged accounts, they may have access to the organization's email, chat, and ticketing systems, and may be able to observe the response as it unfolds. For serious identity incidents, response teams should consider using out-of-band communication channels that do not depend on potentially compromised identities.

Containment, Credential Revocation, Recovery, and Improvement

Containment

The goal of containment is to stop the attacker's use of compromised identities while preserving evidence and minimizing disruption to the business. As the opening story of this chapter illustrated, containment for identity incidents requires more than resetting a password. Effec-

tive containment for a compromised user account typically includes revoking all active sessions and refresh tokens, resetting the password, reviewing and removing any MFA methods the attacker may have registered, reviewing and revoking OAuth application consents, removing mailbox rules and forwarding configurations, and, where the user's device may be compromised, isolating the device. For compromised service accounts and secrets, containment involves rotating the credentials and identifying every system that uses them, so that rotation does not cause an unplanned outage.

Timing matters. Containing a single account too early, before the full scope of the compromise is understood, may alert the attacker and prompt them to use other footholds or accelerate their objectives. Containing too late allows further damage. For serious incidents, I generally recommend identifying as many compromised identities and persistence mechanisms as possible and then removing them in a coordinated action, rather than piecemeal. This is a judgment call that depends on the potential for ongoing harm.

Credential Revocation at Scale

When privileged identities or identity infrastructure have been compromised, containment can require revocation on a much larger scale. This may include resetting all privileged account passwords, rotating service account credentials and secrets, rotating federation signing certificates, invalidating Kerberos tickets across the domain by resetting the accounts used to sign them, and forcing re-authentication for all

users.³ Each of these actions has operational consequences, and some must be performed in a specific sequence to be effective. Organizations that have inventoried their privileged credentials and practiced large-scale rotation, as Chapter 6 recommended, are in a far better position to execute these steps under pressure.

Recovery

In the most severe cases, an organization may conclude that its directory or identity provider can no longer be trusted, because the attacker has had extensive control and may have established persistence that is difficult to discover. Recovery may then require restoring the directory from a known-good backup taken before the compromise, or rebuilding identity infrastructure from scratch.⁴ This is one of the most complex and disruptive operations an IT organization can undertake. Destructive attacks, including ransomware campaigns that target domain controllers, can force the same outcome. In my experience, many organizations have backup procedures for their directories

3. Cybersecurity and Infrastructure Security Agency, "Reset Kerberos Ticket Granting Ticket (KRBTGT) Password Twice, CM0050," *Eviction Strategies Tool*, CISA, <<https://www.cisa.gov/eviction-strategies-tool/info-co-undertakes/CM0050>>, accessed October 22, 2026.

4. Microsoft Corporation, "Active Directory Forest Recovery Procedures," *Microsoft Learn*, <<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-procedures>>, accessed October 22, 2026.

but have never tested a full forest recovery, and they discover gaps in documentation, dependencies, and skills only when they need to perform one.

Post-Incident Improvement

Every identity incident reveals something about the environment. After containment and recovery, organizations should ask which preventive controls failed or were absent, which detections fired and which did not, how long each phase of the response took and why, and which decisions were delayed by unclear responsibilities. The answers should feed into concrete improvements: closing MFA gaps, removing standing privilege, improving telemetry, adding detection rules, updating playbooks, and clarifying decision rights. The organizations that improve most consistently, in my experience, are those that treat these reviews as learning exercises rather than exercises in assigning blame.

REALISTIC THREAT SCENARIO

A healthcare organization's security team receives an alert from its identity provider indicating a risky sign-in for a member of the IT operations team. The analyst reviews the alert and notes that the sign-in used a valid session from an unfamiliar hosting provider. Following an updated identity playbook, the analyst revokes all of the user's sessions and refresh tokens, resets the password, and checks recent identity changes for the account.

The review shows that, in the hour before the alert, the account registered a new authenticator app, granted consent to an unfamiliar application with permission to read all mailboxes, and added a second account to a group with administrative rights

over several cloud subscriptions. The analyst escalates. The IAM team removes the rogue authenticator, revokes the application consent, removes the group membership, and examines the second account, which turns out to be a long-dormant contractor account reactivated the previous day.

The joint investigation traces the initial compromise to a help desk call in which the attacker impersonated the IT operations employee to request an MFA reset. Using telemetry retained for twelve months, the team confirms the scope of access, determines that some mailbox data was read, and notifies the privacy office. Post-incident actions include stronger help desk verification for IT staff, automatic disabling of dormant contractor accounts, and a new detection rule for consent grants requesting mailbox-wide permissions.

The outcome was far better than it might have been, primarily because the playbook covered token revocation and persistence review, the telemetry was available and retained, and the security operations and IAM teams had defined how to work together.

PRACTITIONER CHECKLIST

Which identity sources send telemetry to your detection platform, and which do not? Are token issuance, MFA changes, OAuth consents, and federation changes included?

How long is identity telemetry retained, and is that long enough to investigate an intrusion that began months earlier?

Do you have detections for attacks on identity infrastructure, such as changes to federation trusts, directory replication requests, and new privileged role assignments?

Are break-glass account sign-ins alerted on immediately?

Do your incident playbooks for compromised accounts include revoking sessions and tokens, removing rogue MFA methods, revoking OAuth consents, and removing mailbox rules?

Are decision rights between security operations, IAM, and other teams documented for actions such as disabling executive accounts or rotating federation certificates?

Do you have an out-of-band communication plan for incidents involving compromised identity infrastructure?

Have you tested recovery of your directory and identity provider, including a full forest recovery?

Looking Ahead

This chapter concludes Part II. Together, Chapters 4 through 7 describe the core foundation of identity security: strong authentication, appropriate authorization and governance, protection of privileged access and secrets, and the ability to detect and respond when those defenses are breached. These capabilities have been the focus of identity security programs for many years, and, as I have tried to show, most organizations still have meaningful work to do in each of them.

Part III turns to the challenges that are specific to the age of AI. It begins in Chapter 8 with non-human identities, the service accounts, workloads, bots, APIs, and machine identities that now outnumber human users in many environments. That chapter examines why these identities are so difficult to inventory, govern, and secure, and what practical strategies organizations can use to reduce the risk they represent at scale.

8. Non-Human Identities: The Enterprise Security Challenge

I led an effort to inventory the non-human identities at a large enterprise a few years back. The security team's starting estimate, based on the service accounts registered in the directory, was a few thousand. We began by pulling service principals and application registrations from the cloud identity provider, then roles and access keys from each cloud platform, then Kubernetes service accounts, then API tokens from the CI/CD system, then OAuth applications connected to the major SaaS platforms, then credentials held in the secrets vault, and finally secrets discovered by scanning code repositories.

By the time we stopped counting, the number was many times the original estimate, and we knew the list was still incomplete. More

troubling than the total was what we found when we tried to answer basic questions about each identity. For a large share of them, we could not say who owned it, what it was used for, when its credentials had last changed, or whether anything would break if we disabled it.

I have since seen versions of that exercise play out in many organizations, and the results are remarkably consistent. Non-human identities are numerous, powerful, poorly understood, and largely un-governed. They have accumulated quietly over years of automation, cloud adoption, and integration, and they represent, in my view, one of the largest and least managed areas of identity risk in the modern enterprise. They are also the foundation on which AI agents are being built, which makes addressing them more urgent than ever.

This chapter opens Part III of the book. It examines the different kinds of non-human identities, explains why they are so difficult to inventory, govern, and secure, discusses ownership, lifecycle management, credential rotation, and accountability, describes how workload identity federation can reduce dependence on long-lived secrets, and offers practical strategies for discovering and reducing non-human identity risk at scale.

Understanding Non-Human Identities

A non-human identity is any identity that authenticates to a system and acts within it without being a person. The term covers a wide and varied population. Chapter 6 discussed the credentials many of these identities use. Here, I want to focus on the identities themselves, because effective governance begins with understanding what they are and how they behave.

Service Accounts

Service accounts are the oldest and most familiar category. In on-premises environments, they are typically directory accounts used by applications, scheduled tasks, and services to authenticate to databases, file shares, and other systems. They often look identical to user accounts in the directory, which is part of the problem. Without naming conventions, attributes, or tags that distinguish them, they are easily confused with human accounts and are frequently excluded from, or mishandled by, governance processes designed for people.

Workload Identities

Workload identities represent applications, services, virtual machines, containers, and serverless functions in cloud and cloud-native environments. Cloud platforms provide mechanisms such as managed identities, instance roles, and service principals that allow workloads to authenticate to platform services. Container orchestration platforms such as Kubernetes assign service accounts to workloads running in clusters.¹ These identities are often short-lived in the sense that the workloads they represent may be created and destroyed continuously, but the roles and permissions associated with them may persist for years.

1. The Kubernetes Authors, "Service Accounts," *Kubernetes Documentation*, The Linux Foundation, <<https://kubernetes.io/docs/concepts/security/service-accounts/>>, accessed October 22, 2026.

Application Integrations and OAuth Applications

When one SaaS application connects to another, or when a third-party tool connects to the organization's email, file storage, or CRM platform, the connection is usually represented by an OAuth application or service principal holding tokens and scoped permissions. These integrations are often created by business users or administrators through consent screens, may hold broad access to sensitive data, and are frequently invisible to the central security team. Publicly reported incidents have shown attackers stealing OAuth tokens held by a third-party integration provider and using them to access data across many of that provider's customers, a form of supply chain attack that bypasses the customers' own authentication controls entirely.²

API Clients and Keys

APIs are the connective tissue of modern applications, and API clients authenticate using keys, tokens, client credentials, or certificates. Internal APIs may be consumed by dozens of services. External APIs, including those of payment processors, messaging services, data providers, and AI model providers, are typically accessed using keys

-
2. Austin Larsen, Matt Lin, Tyler McLellan, and Omar ElAhdan, "Widespread Data Theft Targets Salesforce Instances via Salesloft Drift," *Google Cloud Blog*, Google Threat Intelligence Group, August 26, 2025, <<https://cloud.google.com/blog/topics/threat-intelligence/data-theft-salesforce-instances-via-salesloft-drift>>, accessed October 22, 2026.

issued to the organization, which may be shared across teams and environments.

Bots and Automation

Robotic process automation bots, chat bots, scripts, and scheduled jobs perform tasks that once required people. RPA bots in particular often sign in to applications using credentials that resemble user accounts, because the applications they automate were designed for human interaction. CI/CD pipelines are another important category. They frequently hold some of the most powerful credentials in the environment, because they need to build, test, and deploy software into production across many systems.

Devices and Machine Identities

Servers, network devices, IoT and operational technology devices, and endpoints authenticate using certificates, keys, and device identities. In industries such as manufacturing, healthcare, and energy, the number of connected devices can be very large, and many of them have limited ability to support modern authentication or credential rotation.

AI Agents

AI agents, the subject of Chapter 9, are the newest category. They share many characteristics with other non-human identities, since they authenticate using tokens, keys, or workload identities, but they differ in important ways: their behavior is not fully predetermined, they often act on behalf of specific people, and they may choose which tools and data to use. Any weaknesses in how an organization

manages its existing non-human identities are likely to be inherited, and possibly amplified, by its AI agents.

ENTERPRISE FIELD NOTES

One of the most useful questions I ask during non-human identity assessments is not how many identities exist but how many have a named human owner who could be contacted today. In my experience, the proportion is often surprisingly low. Owners leave the company, change roles, or were never recorded in the first place. Identities created by former contractors, by vendors during implementation projects, or by automation scripts are especially likely to be orphaned. An orphaned identity with broad privileges and long-lived credentials is, from an attacker's perspective, close to ideal: powerful, rarely monitored, and unlikely to be noticed if misused.

Why Non-Human Identities Are Difficult to Inventory, Govern, and Secure

If non-human identities are so important, why do organizations manage them so poorly? In my experience, the difficulty stems from several structural factors rather than from any lack of awareness.

The first is fragmentation. Human identities typically flow from an authoritative source, the HR system, into a primary directory and identity provider. Non-human identities have no equivalent source of truth. They are created in many places by many teams: in the directory by infrastructure administrators, in cloud platforms by engineers and infrastructure-as-code templates, in SaaS applications by business administrators, in CI/CD systems by developers, and in third-party tools

by vendors. Each platform has its own identity model, terminology, and administrative interface. There is rarely a single place to look.

The second is scale and velocity. Cloud-native architectures and automation can create and destroy identities continuously. An inventory that was accurate last month may already be out of date. Manual processes cannot keep pace.

The third is the mismatch with human-centric governance. Identity governance processes assume that each identity has a manager who can approve access, a lifecycle tied to employment events, and the ability to respond to review requests. None of these applies naturally to a service account or workload. As a result, non-human identities are often excluded from governance entirely, or included in a way that produces meaningless reviews.

The fourth is operational fragility. Non-human identities are frequently embedded in production processes. Changing their permissions, rotating their credentials, or disabling them can break business-critical systems, sometimes in ways that are difficult to diagnose. That risk makes teams reluctant to touch them, and the reluctance allows weaknesses to persist indefinitely.

The fifth is weak authentication. Most non-human identities cannot use multifactor authentication in any conventional sense. They authenticate using a single secret or key, which means that possession of that credential is sufficient to assume the identity. Many of these credentials are long-lived and stored in places that are difficult to protect.

The sixth is limited visibility into behavior. Human sign-ins generate rich authentication telemetry with device, location, and risk context. Non-human identity activity is often logged less consistently, spread across platforms, and harder to baseline. Detecting misuse

is correspondingly harder, which is one reason attackers value these identities.

Industry groups have begun to formalize these concerns. OWASP, for example, has published guidance identifying the most significant risks associated with non-human identities, including improper off-boarding, secret leakage, overprivileged identities, long-lived secrets, and the reuse of a single identity across multiple purposes.³ In my experience, those risks align closely with what organizations discover when they look closely at their own environments.

Ownership, Lifecycle Management, Rotation, and Accountability

Establishing Ownership

Accountability for non-human identities begins with ownership. Every non-human identity should have an owning team and, ideally, a named individual responsible for it. Ownership should be recorded in a system that is maintained as people change roles, rather than in a spreadsheet that is updated once and forgotten. When an owner leaves the organization, ownership should transfer automatically to their manager or team, and the transfer should trigger a review of whether the identity is still needed.

3. OWASP Foundation, *OWASP Non-Human Identities Top 10, 2025*, OWASP, <<https://owasp.org/www-project-non-human-identities-top-10/2025/top-10-2025/>>, accessed October 22, 2026.

In practice, I have found that assigning ownership to teams or applications rather than only to individuals makes it more durable. A service account that belongs to the payments application, which in turn belongs to the payments engineering team, is less likely to become orphaned than one that belongs to a single engineer.

Lifecycle Management

Non-human identities need lifecycle management just as human identities do, but the lifecycle events are different. Instead of hiring, transfer, and termination, the relevant events include creation of a new application or integration, changes to its function or the systems it connects to, transfer of the application to a new team, decommissioning of the application, and expiry of a vendor contract. Each of these events should trigger corresponding identity actions: provisioning with appropriate scope, adjusting permissions, transferring ownership, and removing the identity and its credentials when it is no longer needed.

Tying non-human identities to the lifecycle of the applications and infrastructure they serve is, in my experience, the most reliable way to prevent orphaned identities. When identities are created through infrastructure-as-code alongside the resources that use them, they can also be removed automatically when those resources are torn down. Identities created manually, outside these pipelines, are the ones most likely to persist long after their purpose has ended.

A related practice is setting expiry dates. Integrations, vendor access, and project-specific identities should have defined end dates, after which they are disabled unless the owner actively confirms they are still needed. Requiring periodic reaffirmation reverses the default from persistence to expiry.

Credential Rotation

Chapter 6 discussed rotation mechanics in detail. From a governance perspective, the key point is that every non-human identity should have a defined credential policy: what type of credential it uses, where that credential is stored, how often it is rotated, and who is responsible for ensuring rotation happens. Identities that cannot be rotated without manual effort and risk should be flagged as candidates for migration to managed identities, workload identity federation, or other mechanisms that eliminate static secrets.

Accountability for Actions

Accountability also means being able to determine what a non-human identity did and on whose behalf. When multiple applications or people share a single service account, it becomes difficult or impossible to attribute actions to a specific source. When an integration acts on behalf of a user, logs should ideally capture both the integration and the user. Avoiding shared identities, using a distinct identity for each workload and purpose, and ensuring that delegated actions are traceable to the delegating party all strengthen accountability. These principles will become especially important for AI agents, as Chapter 9 explains.

BEST PRACTICES THAT WORK

Define a minimum metadata standard for every non-human identity and enforce it at creation. In practice, this tends to include the owning team and named owner, the application or service it supports, its purpose, the environment it operates in,

the type of credential it uses, its expiry or review date, and its sensitivity or privilege level. Where identities are created through infrastructure-as-code or a provisioning workflow, these fields can be required before creation succeeds. Identities that lack this metadata should be treated as unknown and prioritized for investigation. This sounds administrative, and it is, but without it nearly every other control for non-human identities becomes guesswork.

Workload Identity Federation and Reducing Dependence on Long-Lived Secrets

The most promising direction for securing non-human identities is to remove long-lived secrets from the equation wherever possible. Instead of giving a workload a static key that it must store and protect, the workload proves its identity based on where and how it is running, and receives short-lived credentials in return.

Managed and Platform-Native Identities

Within a single cloud platform, the simplest approach is to use the platform's native workload identity mechanisms. A virtual machine, container, or serverless function can be assigned an identity by the platform, and the platform provides short-lived tokens to the workload automatically. No secret is stored in code or configuration, and credentials rotate continuously without any action from the application team. Where these mechanisms are available, I would generally recommend them over static access keys.

Workload Identity Federation

Many workloads, however, need to access resources outside the environment in which they run. A CI/CD pipeline hosted by one provider may need to deploy into a cloud platform operated by another. An application running on premises may need to access cloud services. A workload in one cloud may need resources in another.

Workload identity federation addresses these scenarios using the same trust principles that federation provides for human users. The workload obtains a signed token from the identity provider of the environment in which it runs, typically in the form of an OpenID Connect token that describes the workload, such as the repository, branch, and pipeline that produced it. The target environment is configured to trust that identity provider and to accept tokens that match specific conditions. When the workload presents its token, the target environment exchanges it for short-lived credentials scoped to the permissions configured for that workload.

The security advantages are significant. There is no long-lived secret to steal, leak, or rotate. Trust can be scoped narrowly, for example to a specific repository and branch, so that only the intended pipeline can obtain credentials. And credentials expire quickly, limiting the value of any that are intercepted. Major cloud platforms and CI/CD providers now support this pattern, and in my experience, migrating

pipeline credentials to federation is one of the highest-value improvements an organization can make for software supply chain security.⁴

There are cautions. Federation shifts trust to the configuration of the trust relationship itself. If the conditions are written too broadly, for example trusting any workload from a given provider rather than a specific repository, an attacker who can run a workload on that provider may be able to obtain credentials. Trust configurations deserve careful review and monitoring, just as federation trusts for human identities do.

Standards for Workload Identity

Beyond cloud-specific mechanisms, open standards are emerging to provide consistent workload identity across heterogeneous environments. The SPIFFE specification, with its reference implementation SPIRE, defines a standard way to issue cryptographically verifiable identities to workloads based on attestation of their runtime proper-

4. GitHub, Inc., "About Security Hardening with OpenID Connect," *GitHub Docs*, <<https://docs.github.com/en/actions/security-for-github-actions/security-hardening-your-deployments/about-security-hardening-with-openid-connect>>, accessed October 22, 2026.

ties, and to rotate those identities automatically.⁵ Service mesh technologies often use such identities to authenticate and encrypt communication between services. For organizations operating complex, multi-cloud, or hybrid architectures, these standards may offer a path toward a unified model of workload identity, although adoption requires meaningful engineering investment.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Static secrets in a vault, or federated short-lived credentials? Vaulting static secrets is broadly compatible, relatively quick to implement, and works with legacy applications and third-party services that accept only keys. It centralizes control and visibility but leaves long-lived credentials in existence and requires ongoing rotation.

Federation and platform-native identities eliminate long-lived secrets and rotate credentials automatically, but they require support from both the workload environment and the target system, may require application changes, and shift risk to the correctness of trust configurations.

In practice, most enterprises will operate both for years. A sensible strategy is to make federation and native identities the default for all new workloads and pipelines, migrate the highest-privilege existing static credentials first, and reserve vaulted

5. Cloud Native Computing Foundation, "SPIFFE and SPIRE Projects Graduate from Cloud Native Computing Foundation Incubator," press release, September 20, 2022, <<https://www.cncf.io/announcements/2022/09/20/spiffe-and-spire-projects-graduate-from-cloud-native-computing-foundation-incubator/>>, accessed October 22, 2026.

static secrets for systems that cannot support anything better, with rotation enforced and monitored.

Practical Strategies for Discovering and Reducing Non-Human Identity Risk at Scale

Given the scale of the problem, organizations need a deliberate, phased approach. The following sequence reflects what I have seen work in large environments.

Discover

Begin by building an inventory across all of the places non-human identities live: directories, cloud identity providers, each cloud platform's IAM system, container platforms, CI/CD systems, secrets managers, SaaS platforms and their connected applications, API gateways, and certificate management systems. Supplement platform data with scanning for secrets in code, configuration, and collaboration tools, since those often reveal identities that no platform inventory captures. Specialized non-human identity security tools can automate much of this discovery, correlating identities across platforms and mapping their permissions and usage. Whatever tools are used, the goal is a continuously updated inventory, not a one-time spreadsheet.

Contextualize

For each identity, establish its owner, purpose, permissions, credential type and age, and recent activity. Usage data is particularly valuable. An identity that has not authenticated in many months, or a permis-

sion that has never been exercised, is a strong candidate for removal. Mapping the relationships between identities, such as which service principal can assume which role, or which pipeline can access which secrets, reveals escalation paths that individual records do not.

Prioritize

Focus first on identities that combine high privilege with high exposure: identities with administrative or broad data access, long-lived credentials, credentials found in code or shared locations, no identified owner, access from third-party integrations, or permissions that allow escalation to tier-zero systems. A relatively small number of identities often account for a disproportionate share of risk.

Remediate

Remediation follows the patterns discussed in Chapters 5 and 6, adapted for non-human identities. Remove dormant identities, starting by disabling rather than deleting them so they can be restored if needed. Reduce permissions to what each identity actually uses. Split shared identities into distinct identities for each purpose. Rotate exposed and aged credentials. Migrate high-value static credentials to managed identities or federation. Prevent service accounts from signing in interactively. Review third-party integrations and revoke those that are unused or excessively privileged. Because of the operational fragility discussed earlier, coordinate remediation with owners, communicate changes in advance, and monitor closely for breakage.

Govern and Monitor

Finally, put durable controls in place so the problem does not recur. Require metadata and ownership at creation. Route new non-human identities through provisioning workflows or infrastructure-as-code. Set expiry dates and require reaffirmation. Establish policies for which credential types are acceptable in which contexts. Include non-human identities in access reviews, but design those reviews for owners of applications and integrations rather than for managers of people. And extend identity threat detection to non-human identities, alerting on behavior such as interactive sign-ins by service accounts, use of credentials from unexpected locations, sudden increases in data access, and creation of new credentials for privileged identities.

REALISTIC THREAT SCENARIO

A manufacturing company connects a third-party analytics platform to its CRM and file storage systems through OAuth integrations. The integrations are approved by a business administrator, granted broad read access to customer and sales data, and never reviewed by the security team. Two years later, the analytics vendor is breached, and the attacker obtains the OAuth tokens the vendor holds for its customers.

Using those tokens, the attacker queries the company's CRM and downloads large volumes of customer records and contract documents from file storage. Because the activity comes from a legitimate, previously authorized integration, the company's authentication controls, including its phishing-resistant MFA, play no role. The company learns of the breach only when the vendor notifies its customers several weeks later.

An inventory of third-party integrations with recorded owners, a review that reduced the integration's permissions to the specific objects it needed, restrictions on the network locations from which the integration's tokens could be used where the platform supported it, monitoring for unusual data volumes accessed by integrations, and a periodic reaffirmation requirement that would have revealed the integration was used only occasionally would each have reduced the exposure. The scenario also illustrates that non-human identity risk extends beyond the organization's own boundaries to the vendors it trusts with access.

PRACTITIONER CHECKLIST

Do you have a continuously updated inventory of non-human identities across directories, cloud platforms, container platforms, CI/CD systems, SaaS integrations, secrets managers, and certificate systems?

What proportion of non-human identities has a named, current owner?

Are non-human identities required to carry metadata such as purpose, owner, environment, credential type, and expiry at creation?

How many non-human identities have not authenticated in the past ninety days, and are they being disabled?

Which high-privilege identities still rely on long-lived static credentials, and is there a plan to migrate them to managed identities or federation?

Are federation trust conditions scoped narrowly, for example to specific repositories, branches, and workloads?

Are service accounts prevented from signing in interactively?

Do you inventory and periodically review third-party OAuth integrations and the data they can access?

Does your identity detection program monitor non-human identity behavior as well as human sign-ins?

Looking Ahead

Non-human identities have been accumulating in enterprise environments for decades, and most organizations are only beginning to manage them with the rigor they apply to human users. The arrival of AI agents makes that work considerably more urgent. Agents depend on the same mechanisms, including tokens, keys, workload identities, and OAuth delegations, and they introduce new characteristics that existing approaches were not designed to handle.

Chapter 9 turns to those characteristics directly. It examines how AI agents operate and interact with enterprise systems, how to establish identity for agents, models, tools, and delegated tasks, how to define permissions and boundaries for autonomous actions, how to secure agent-to-agent communication, tool use, and access to sensitive data, and how to monitor agent behavior, maintain accountability, and implement human approval controls where the stakes require them.

9. Securing AI Agents and Autonomous Systems

Not long ago, I sat in on a demonstration that a business unit had arranged for its leadership team. An engineer had built an AI agent to help the sales organization prepare for customer meetings. Given a customer name, the agent would search the CRM for account history, read recent email threads with the customer, pull relevant contracts from the document repository, summarize open support tickets, and draft a briefing. It could also, if asked, send a follow-up email or schedule a meeting. The demonstration was impressive, and the room was enthusiastic. When the presentation ended, I asked a few questions. Whose credentials did the agent use to access the CRM and the mailbox? Could it read every customer's data, or only the accounts belonging to the person using it? What would happen if a customer's email contained instructions telling the agent to forward contracts to an external address? Who would know if it did? The engineer

answered honestly. The agent used a service account with broad read access across the CRM and file repository, and delegated access to the user's mailbox. Nobody had considered the email scenario. Logging consisted of the application's debug output.

I do not tell this story to criticize the engineer, who had built exactly what was asked, quickly and well. I tell it because I believe it represents the current state of AI agent adoption in many enterprises. Agents are being built and deployed by people focused on productivity and capability, using identity and access patterns borrowed from earlier generations of automation. Those patterns were not designed for systems that interpret natural language, choose their own actions, and can be influenced by the content they process.

This chapter examines how to secure AI agents from an identity perspective. It explains how agents operate and interact with enterprise systems, how to establish identity for agents, models, tools, and delegated tasks, how to define permissions and boundaries for autonomous actions, how to secure agent-to-agent communication, tool use, and access to sensitive data, and how to monitor agent behavior, maintain accountability, and implement human approval controls. Much of the technology in this area is evolving rapidly, and I expect specific standards and products to change. The underlying principles, however, are likely to remain relevant.

How AI Agents Operate and Interact With Enterprise Systems

An AI agent typically combines several components. A language model interprets instructions, reasons about how to accomplish a goal, and decides what to do next. An orchestration layer manages the loop of planning, acting, and observing results. Tools give the

agent the ability to act: querying a database, calling an API, reading a file, searching the web, sending an email, running code, or invoking another agent. Context includes the user's instructions, system instructions set by the developer, retrieved documents, the results of previous tool calls, and in some cases memory carried across sessions.

From an identity perspective, the critical observation is that tools are where the agent touches enterprise systems, and every tool call is an authenticated, authorized request. When an agent reads a mailbox, it presents a token. When it queries the CRM, it presents credentials. The agent's capabilities are therefore defined not only by what the model can reason about but by what identities and permissions its tools carry.

The ecosystem for connecting agents to tools is maturing quickly. The Model Context Protocol, introduced in late 2024 and since adopted widely across the industry, provides a standard way for agents to discover and invoke tools and data sources exposed by servers.¹ Agent-to-agent protocols, such as the A2A protocol introduced in 2025, aim to standardize how agents built on different platforms discover one another and delegate tasks.² Major enterprise software vendors are embedding agents directly into their platforms, and cloud

1. Anthropic, "Introducing the Model Context Protocol," *Anthropic News*, November 25, 2024, <<https://www.anthropic.com/news/model-context-protocol>>, accessed October 22, 2026.

2. Rao Surapaneni, Miku Jha, Michael Vakoc, and Todd Segal, "Announcing the Agent2Agent Protocol (A2A)," *Google for Developers Blog*, April 9, 2025, <<https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interopability/>>, accessed October 22, 2026.

providers offer frameworks and runtimes for building custom agents. In practice, this means that an enterprise may soon host agents built by its own developers, agents embedded in its SaaS applications, agents provided by third parties, and agents adopted informally by individual employees, all interacting with the same systems and data.

Several characteristics distinguish agents from the non-human identities discussed in Chapter 8, and each has implications for identity security.

Agents are non-deterministic. The same instructions may lead to different sequences of actions on different occasions. Permissions cannot be scoped by predicting exactly what the agent will do, because that is not fully knowable in advance.

Agents often act on behalf of people. A user asks an agent to accomplish something, and the agent uses some combination of its own identity and the user's delegated authority to do it. This blurs the line between human and machine identity in ways that existing logging and authorization models do not always capture.

Agents process untrusted content. Emails, documents, web pages, support tickets, and tool outputs all enter the agent's context. If that content contains instructions, the model may follow them. This vulnerability, known as prompt injection, means that anyone who can place content in front of an agent may be able to influence its behavior, and through it, the identities and permissions it holds.

Agents can chain actions across systems quickly. An agent may retrieve data from one system and send it through another within seconds, at a scale and speed that human review cannot match.

ENTERPRISE FIELD NOTES

In discussions with security leaders, I have found that one of the most common early mistakes is evaluating agent risk pri-

marily through the lens of the model: which provider, which version, whether it hallucinates. Those questions matter, but from a security perspective the more consequential questions are about access. Which systems can the agent reach? With whose authority? What can it change, delete, or send? An agent built on an excellent model with excessive permissions may represent far greater risk than one built on a weaker model with narrowly scoped, read-only access.

Establishing Identity for Agents, Models, Tools, and Delegated Tasks

The first principle I would offer is simple: every agent should have its own identity. Agents should not run under shared service accounts, borrowed human credentials, or generic API keys used by many applications. A distinct identity for each agent, and ideally for each deployment or instance where the distinction matters, makes it possible to scope permissions, attribute actions, monitor behavior, and revoke access precisely.

The Agent Registry

Distinct identities are of limited value without an inventory that describes them. I recommend that organizations maintain an agent registry, an extension of the non-human identity inventory described in Chapter 8, that records each agent's purpose, owner, model and version, the tools and data sources it can access, the identities and permissions it uses, whether it acts on behalf of users, the actions it can take autonomously and those that require approval, and its approval

and review status. Several identity and security vendors now offer capabilities specifically designed for registering and governing agents, and some identity providers have begun to treat agents as a distinct identity type. Whatever tools are used, the registry should become the authoritative answer to the question of which agents exist and what they are allowed to do.

Delegation and On-Behalf-Of Access

When an agent acts on behalf of a user, there are broadly three patterns for how it obtains authority.

In the first, the agent uses the user's own credentials or tokens directly, effectively impersonating the user. This is simple but problematic. Systems cannot distinguish the agent's actions from the user's, the agent inherits all of the user's permissions rather than only those needed for the task, and the user's credentials are exposed to the agent's runtime.

In the second, the agent uses its own identity with broad permissions and applies its own logic to decide what each user should be allowed to see. This is the pattern in the demonstration I described at the start of this chapter. It concentrates enormous access in a single identity and relies on the agent, which may be manipulated, to enforce authorization correctly. It also creates a classic confused deputy problem, in which a user or attacker may be able to induce the agent to use its broad privileges on their behalf.³

3. Norm Hardy, "The Confused Deputy (or Why Capabilities Might Have Been Invented)," *ACM SIGOPS Operating Systems Review*, Vol. 22, No. 4, October 1988, pp. 36-38.

In the third, which I generally recommend, the agent acts with delegated authority that combines both identities. Using standards such as OAuth 2.0 token exchange, the agent obtains a token that represents the user, carries an indication that the agent is the actor, and is scoped to the specific permissions the task requires.⁴ Downstream systems enforce authorization based on the user's entitlements, intersected with the agent's allowed scope, and their logs can record both the user and the agent. The effective permission becomes the overlap between what the user may do and what the agent is permitted to do, rather than the union of the two.

Standards and product support for this pattern are still maturing, particularly across multi-step chains in which one agent delegates to another, which delegates to a tool. The OAuth community and protocol bodies have active work under way on agent authorization, including the authorization model adopted for the Model Context Protocol.⁵ I would encourage architects to design toward this delegated model now, even where current tooling requires compromises.

4. Michael B. Jones, Anthony Nadalin, Brian Campbell, John Bradley, and Chuck Mortimore, *OAuth 2.0 Token Exchange*, RFC 8693, Internet Engineering Task Force, January 2020, <<https://www.rfc-editor.org/rfc/rfc8693>>, accessed October 22, 2026.

5. Model Context Protocol, "Authorization," *Model Context Protocol Specification*, Version 2026-07-28, <<https://modelcontextprotocol.io/specification/2026-07-28/basic/authorization>>, accessed October 22, 2026.

Identity for Tools and Models

Tools and model endpoints need identities too. A tool server exposed to agents should authenticate the agents that call it and enforce its own authorization, rather than trusting any caller. Agents should verify that they are connecting to legitimate tool servers, since a malicious or compromised tool can return manipulated data or capture the credentials and context passed to it. Access to model APIs, often through keys issued by external providers, should itself be treated as a non-human identity with an owner, scope, and rotation policy, since stolen model API keys can be abused for cost, data access, or to impersonate the organization's applications.

Defining Permissions and Boundaries for Autonomous Actions

Because agent behavior cannot be fully predicted, permissions must be designed around boundaries rather than expected actions. The question is not what the agent will do, but what is the worst it could do with the access it has, and is that acceptable?

Least Agency

Security practitioners have begun using the term excessive agency to describe the risk that an agent has more capability, permissions, or autonomy than its task requires, and OWASP includes it among the

leading risks for applications built on large language models.⁶ The corresponding principle, which some call least agency, extends least privilege to cover not only permissions but the range of tools available and the degree of autonomy granted. An agent that summarizes documents does not need the ability to send email. An agent that drafts responses does not need to send them without review. An agent that reads a database does not need write access.

In practice, this means limiting the tools each agent can call to those essential for its purpose; scoping each tool's permissions narrowly, for example to specific records, folders, or customers; preferring read-only access wherever possible; separating high-risk capabilities, such as sending external communications, moving money, deleting data, or changing permissions, into distinct tools that require additional controls; and setting rate and volume limits so that an agent cannot retrieve or modify unusually large amounts of data in a short period.

Time and Context Boundaries

The just-in-time principles discussed in Chapters 5 and 6 apply naturally to agents. Rather than holding standing permissions, an agent can receive credentials scoped to a specific task and valid only for its duration. When the task ends, the credentials expire. Contextual conditions, such as the user who initiated the task, the data classifi-

6. OWASP Foundation, "LLM06:2025 Excessive Agency," *OWASP Top 10 for LLM Applications 2025*, OWASP GenAI Security Project, <<https://genai.owasp.org/llmrisk/llm062025-excessive-agency/>>, accessed October 22, 2026.

cation involved, and the environment, can further constrain what the agent may do. Policy engines that externalize authorization decisions are particularly well suited to this purpose, because they allow the organization to define and audit agent permissions centrally rather than embedding them in each agent's code.

Containing the Effects of Prompt Injection

Prompt injection cannot currently be prevented reliably at the model level. Model providers and security vendors have developed mitigations, including input filtering, instruction hierarchies, and classifiers that attempt to detect injected instructions, and these reduce the risk. In my view, however, organizations should design on the assumption that a determined attacker may sometimes succeed in influencing an agent's behavior, and should rely on identity and authorization boundaries to limit what a manipulated agent can do.

A useful way to reason about this is to look for dangerous combinations. An agent that has access to sensitive data, processes content from untrusted sources, and has the ability to communicate externally can potentially be manipulated into exfiltrating that data.⁷ Removing any one of those three properties for a given agent, or placing strong controls such as human approval on the external communication

7. Simon Willison, "The Lethal Trifecta for AI Agents: Private Data, Untrusted Content, and External Communication," *Simon Willison's Weblog*, June 16, 2025, <<https://simonwillis.net/2025/Jun/16/the-lethal-trifecta/>>, accessed October 22, 2026.

path, significantly reduces the risk. Many of the agent designs I review combine all three without realizing it.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Broad agent identity with application-level filtering, or delegated user-scoped access? Granting an agent a broad service identity and relying on the agent's own logic to filter what each user sees is faster to build, works with systems that do not support delegation, and avoids complex token flows. It also concentrates access in one identity, makes the agent a high-value target, and depends on the agent correctly enforcing authorization even when it is being manipulated.

Delegated, user-scoped access ensures that downstream systems enforce the user's own entitlements, limits the effect of a manipulated agent to what that user could already do, and produces clearer audit trails. It requires support for delegation in the identity provider and target systems, and it can be more complex to implement, particularly for multi-agent chains.

For agents that access sensitive data on behalf of many different users, I would treat delegated access as the target architecture and accept broad service identities only for narrowly scoped, low-sensitivity use cases or as a documented interim measure.

Securing Agent-to-Agent Communication, Tool Use, and Sensitive Data

Agent-to-Agent Communication

As multi-agent systems become more common, agents will increasingly delegate tasks to other agents, some within the organization and some operated by partners or vendors. Each such interaction is a trust decision. The calling agent should authenticate the agent it is delegating to. The receiving agent should authenticate the caller and determine what authority, if any, has been delegated to it. Delegation chains should preserve the identity of the originating user and each intermediate agent, so that authorization at the end of the chain reflects the full context and logs record who initiated what. Permissions should narrow, not widen, as tasks move along the chain. And content returned from another agent should be treated as untrusted input, since a compromised or manipulated agent can pass injected instructions downstream.

Tool Use

Tool servers and connectors represent a significant new attack surface. Organizations should maintain an approved catalog of tools and tool servers that agents may use, review new tools before they are connected, and be cautious about tools obtained from public sources without vetting. Tool descriptions and metadata can themselves contain injected instructions, so tool registration should be controlled. Credentials used by tools should follow the principles in Chapter 6:

stored in vaults or replaced by short-lived, federated credentials, never embedded in agent configuration or prompts.

Access to Sensitive Data

Agents that retrieve data, whether by querying systems directly or through retrieval-augmented generation from indexed document collections, must respect the same access controls that apply to the people they serve. A common failure mode is building a search index or knowledge base from documents with mixed permissions and then allowing an agent to retrieve from it without enforcing per-user access, effectively exposing restricted content to anyone who asks the right question. Retrieval systems should carry document permissions into the index and enforce them at query time based on the requesting user's identity. Data classification and data loss prevention controls should extend to agent outputs, including content the agent sends externally or writes into other systems. Organizations should also consider what data agents send to external model providers and whether contractual and technical protections are adequate for its sensitivity.

BEST PRACTICES THAT WORK

Before approving an agent for production, conduct a short, structured review focused on identity and access rather than model quality. In practice, this tends to include confirming that the agent has its own registered identity and named owner; listing every tool it can call and the permissions behind each one; confirming whether it acts on behalf of users and how that delegation is implemented; identifying whether it combines access to sensitive data, exposure to untrusted content, and the ability to communicate externally; defining which actions require human approval; confirming that its actions are logged with both agent and user identity; and verifying that it can be disabled

quickly. Many organizations already perform similar reviews for new applications and integrations. Extending them to agents is a modest step that catches a large share of design flaws.

Monitoring Agent Behavior, Accountability, and Human Approval Controls

Monitoring

Agents should generate telemetry that supports security monitoring and investigation, not merely debugging. At a minimum, logs should capture the agent's identity, the user on whose behalf it acted, the task or instruction it received, each tool call it made with parameters and results, the data it accessed, and the actions it took in external systems. Downstream systems should log agent actions in a way that identifies the agent as the actor. Privacy and data retention considerations apply, since agent logs may contain sensitive content, but the absence of usable logs makes investigation nearly impossible.

Behavioral monitoring for agents can build on the ITDR approaches in Chapter 7. Useful signals include tool calls the agent has never made before, access to data outside its normal scope, unusually high volumes of retrieval or actions, attempts to call tools it is not authorized to use, external communications to new destinations, and patterns that suggest the agent is following instructions embedded in content rather than from its user. Because agent behavior varies naturally, baselines will be noisier than for traditional service accounts, and I would expect detection techniques in this area to mature significantly over the next few years.

Accountability

When an agent takes a harmful action, the organization needs to be able to determine what happened, why, and who is responsible. Technical accountability depends on the identity design and logging described above. Organizational accountability depends on clear ownership. Every agent should have an owner responsible for its configuration, permissions, and behavior, and every user who initiates agent actions should understand that they remain accountable for what is done on their behalf. In my view, agents should be treated as tools wielded by people and teams, not as independent actors that absorb responsibility for outcomes.

Human Approval Controls

Human-in-the-loop approval is one of the most effective controls for limiting the impact of agent errors and manipulation, but it must be designed carefully. If every action requires approval, users will experience the same fatigue described in Chapter 4 for MFA prompts and will approve without reading. If no actions require approval, high-impact mistakes proceed unchecked.

A risk-tiered approach tends to work best. Low-risk, reversible, read-only actions can proceed autonomously. Moderate-risk actions can proceed with notification and easy reversal. High-risk actions, such as sending communications to external parties, executing financial transactions, deleting data, changing permissions or security configurations, or acting on data above a defined sensitivity, should require explicit human approval. The approval request should show clearly what the agent intends to do, including the full content of mes-

sages and the specific records affected, so that the approver can make an informed decision. Approval should be granted by an appropriate person, which for some actions may be someone other than the user who initiated the task.

Finally, organizations need the ability to stop agents quickly. This includes disabling an individual agent's identity, revoking its tokens, suspending specific tools across all agents, and halting agent activity broadly if a systemic problem is discovered. These capabilities should be tested before they are needed.

REALISTIC THREAT SCENARIO

A company deploys an AI agent to help its procurement team process supplier correspondence. The agent reads incoming email in a shared procurement mailbox, extracts invoice details, updates records in the procurement system, and drafts replies. To simplify implementation, it runs under a service identity with full access to the mailbox, read and write access to supplier records, and permission to send email.

An attacker sends an email that appears to be a routine invoice from a real supplier. Hidden in the message is text instructing the agent to update the supplier's banking details to a new account and to reply confirming the change. The agent processes the email, follows the embedded instructions, updates the supplier record, and sends a confirmation. The next payment run sends funds to the attacker.

The agent was not compromised in the traditional sense. Its credentials were never stolen. It was manipulated into using its legitimate access on the attacker's behalf.

Several identity and authorization controls would have prevented the loss: separating the ability to change banking details

into a distinct, high-risk tool requiring human approval with an out-of-band verification; restricting the agent's write access to invoice fields only; removing its ability to send external email without review; monitoring for agent-initiated changes to payment details; and logging that clearly showed the agent, rather than a staff member, made the change. This scenario closely mirrors the business email compromise patterns described in Chapter 2, with an agent standing in for the deceived employee.

PRACTITIONER CHECKLIST

Do you maintain a registry of AI agents operating in your environment, including those embedded in SaaS platforms and those adopted by individual employees?

Does each agent have its own identity and a named owner?

For agents acting on behalf of users, is access delegated and scoped to the user's entitlements, or does the agent use a broad service identity?

Which agents combine access to sensitive data, exposure to untrusted content, and the ability to communicate externally?

Are high-impact actions separated into distinct tools that require human approval?

Do retrieval systems enforce per-user permissions on the documents they index?

Are tool servers and connectors reviewed and approved before agents may use them?

Do logs record both the agent and the user for every action, with enough detail to support investigation?

Can you disable an agent, revoke its tokens, or suspend a tool across all agents quickly, and has that capability been tested?

Looking Ahead

This chapter has focused on securing the agents organizations deploy themselves. Attackers, of course, also have access to AI, and they are using it to make identity-based attacks more convincing, more scalable, and more difficult to detect.

Chapter 10 examines those threats. It discusses AI-enabled phishing, impersonation, deepfakes, and social engineering; automated credential attacks and large-scale identity exploitation; synthetic identities and the manipulation of identity verification; attacks against identity infrastructure, authentication workflows, and recovery processes; and how to approach threat modeling for AI-era attacks and prioritize defensive controls accordingly.

10. AI-Powered Attacks and Emerging Identity Threats

Not long ago, a colleague who leads security at a large company played me a voicemail. It was from the company's chief financial officer, asking the treasury team to expedite a payment to a new vendor ahead of an acquisition announcement and to keep the matter confidential. The voice was unmistakably the CFO's: the same cadence, the same slight hesitation before numbers, the same phrase he used to close calls. The treasury analyst who received it had been suspicious only because the request arrived on a weekend and bypassed the normal approval workflow. When she called the CFO on his known mobile number, he had no idea what she was talking about. The voicemail had been generated from recordings of his public earnings calls and conference presentations.

My colleague told me that what unsettled him most was not that the attack had been attempted. Executive impersonation is an old

fraud. What unsettled him was that his team's security awareness training had, for years, taught employees to verify unusual requests by listening for something that seemed off. In this case, nothing about the voice seemed off at all. The analyst's instinct to verify through a known channel, rather than her ability to detect a fake, was what stopped the fraud.

I believe that story captures the central challenge of this chapter. AI is not creating entirely new categories of identity attack. Phishing, impersonation, credential theft, fraud, and manipulation of recovery processes have existed for decades. What AI changes is their quality, their scale, their cost, and the degree to which human judgment can be relied on to detect them. Many of the signals that people and processes have used to establish trust are becoming easier to fake.

This chapter examines AI-enabled phishing, impersonation, deep-fakes, and social engineering; automated credential attacks and large-scale identity exploitation; synthetic identities, identity fraud, and the manipulation of identity verification; attacks against identity infrastructure, authentication workflows, and recovery processes; and how organizations can approach threat modeling for AI-era attacks and prioritize their defenses.

AI-Enabled Phishing, Impersonation, Deepfakes, and Social Engineering

Phishing at Scale and With Precision

Chapter 2 noted that generative AI allows attackers to produce fluent, personalized phishing messages at scale. It is worth examining why that matters so much. Historically, attackers faced a trade-off between vol-

ume and quality. Mass phishing campaigns reached many recipients with generic messages that were relatively easy to spot. Targeted spear phishing, researched and written by hand for specific individuals, was far more convincing but labor-intensive. AI substantially erodes that trade-off. An attacker can now gather information about many targets from professional networking sites, company websites, press releases, and previous data breaches, and generate individually tailored messages for each of them, in multiple languages, with appropriate tone and context, at a cost approaching that of mass phishing.

Several research studies and industry experiments have suggested that AI-generated phishing can perform comparably to, and in some cases better than, messages crafted by skilled human operators.¹ I would treat specific success rates with caution, since results vary with methodology and context, but the direction is consistent with what security teams report observing. AI also allows attackers to sustain conversations, responding plausibly to a target's questions over email or chat, which defeats the simple advice to reply and ask for clarification.

1. Fred Heiding, Simon Lermen, Andrew Kao, Bruce Schneier, and Arun Vishwanath, "Evaluating Large Language Models' Capability to Launch Fully Automated Spear Phishing Campaigns: Validated on Human Subjects," arXiv:2412.00586, November 30, 2024, <<https://arxiv.org/abs/2412.00586>>, accessed October 22, 2026.

Voice Cloning and Deepfake Video

Voice cloning has advanced to the point where a convincing imitation can, by many accounts, be produced from a relatively short sample of recorded speech. Executives, public officials, and anyone who speaks publicly or posts videos online provide ample source material. Law enforcement agencies, including the FBI, have issued public warnings about campaigns using AI-generated voice messages to impersonate senior officials and executives, often as a first step toward establishing trust before moving the conversation to another channel.²

Real-time deepfake video is less mature but improving. The widely reported case of an employee who authorized substantial transfers after a video conference in which multiple senior colleagues appear to have been synthetic illustrated that video presence can no longer be treated as proof of identity.³ In my view, any process that relies on recognizing a person's voice or face over a remote channel as the primary means of verifying identity should now be considered weak.

2. Federal Bureau of Investigation, *Senior US Officials Impersonated in Malicious Messaging Campaign*, Public Service Announcement I-051525-PSA, Internet Crime Complaint Center, May 15, 2025, <<https://www.ic3.gov/PSA/2025/PSA250515>>, accessed October 22, 2026.

3. Magramo, "British Engineering Giant Arup Revealed as \$25 Million Deepfake Scam Victim," *CNN Business*, May 16, 2024.

Targeting the People Who Grant Access

AI-enabled social engineering is particularly dangerous when it targets the people and processes that control identity. Help desk agents who reset passwords and MFA registrations, IT administrators who provision access, HR staff who onboard new employees, and finance staff who approve payment changes are all, in effect, identity gatekeepers. An attacker who can convincingly impersonate an employee to the help desk, or an executive to the finance team, can bypass technical controls entirely. Several significant publicly reported intrusions have begun with help desk social engineering, and AI makes the impersonation more convincing and easier to scale.⁴

ENTERPRISE FIELD NOTES

When I review verification procedures for help desks and finance teams, I frequently find that they depend on information an attacker can obtain or infer: an employee ID number, a manager's name, a date of birth, the last four digits of a phone number, or simply the ability to answer questions confidently. These procedures were designed to stop casual impostors, not well-prepared ones. Now that attackers can combine leaked personal data with a cloned voice, I encourage organizations to assume that knowledge-based and voice-based verification can be defeated, and to rely instead on verification that requires possession of an enrolled device or credential, or confirmation through a separate, trusted channel.

4. FBI and CISA, *Scattered Spider*, Joint Cybersecurity Advisory AA23-320A.

Automated Credential Attacks and Large-Scale Identity Exploitation

Credential attacks have been automated for many years. Credential stuffing tools test leaked username and password pairs against many sites, and password spraying tools try common passwords across many accounts. AI adds several capabilities to this established ecosystem.

First, AI can make credential attacks harder to detect. Automated tools can vary timing, source infrastructure, and behavior to resemble legitimate users more closely, and they can be routed through large pools of residential proxies so that traffic appears to come from ordinary consumer internet connections near the target's users. Simple defenses based on blocking known malicious IP addresses or detecting obvious automation become less effective.

Second, AI can improve password guessing. Models trained on leaked password collections can generate likely candidates for specific users based on personal information, organizational naming patterns, and common transformations, potentially increasing the success rate of targeted guessing.

Third, AI can help attackers defeat some anti-automation controls. Many CAPTCHA and bot detection systems were designed around tasks that were difficult for software, such as recognizing distorted text or objects in images. Advances in computer vision and model

capability have made several of these challenges easier for automated systems to solve.⁵

Fourth, and in my view most significantly, AI agents may allow attackers to automate not only the initial credential attack but the steps that follow. An attacker who obtains access to many accounts faces a bottleneck in deciding what to do with each one: exploring mailboxes, identifying valuable data, recognizing opportunities for fraud or escalation. AI can triage compromised accounts at scale, searching for financial information, credentials, sensitive documents, and relationships that can be exploited. At least one major AI developer has publicly reported identifying and disrupting an espionage campaign in which an AI system performed a large share of the intrusion work with limited human direction.⁶ I expect the use of AI to accelerate post-compromise activity to grow, which will shrink the time defenders have to detect and respond.

The implications for defenders reinforce themes from earlier chapters. Phishing-resistant authentication remains the most effective defense against credential theft. Breached password screening helps prevent users from choosing credentials already in attacker collections. Device-bound sessions and conditional access requiring managed de-

5. Andreas Plesner, Tobias Vontobel, and Roger Wattenhofer, "Breaking reCAPTCHA_{v2}," arXiv:2409.08831, September 13, 2024, <<https://arxiv.org/abs/2409.08831>>, accessed October 22, 2026.

6. Anthropic, "Disrupting the First Reported AI-Orchestrated Cyber Espionage Campaign," *Anthropic News*, November 13, 2025, <<https://www.anthropic.com/news/disrupting-AI-espionage>>, accessed October 22, 2026.

vices reduce the value of stolen credentials. And identity threat detection must be capable of recognizing compromised accounts quickly, because the window between compromise and damage is likely to narrow.

Synthetic Identities, Identity Fraud, and the Manipulation of Identity Verification

Synthetic Identities

A synthetic identity is a fabricated identity, often combining real and invented personal information, created to pass verification checks. Synthetic identity fraud has long affected financial services, where criminals build credit histories for fictitious people and then exploit them. AI makes synthetic identities easier to create and more convincing. Generative models can produce realistic photographs of people who do not exist, fabricate supporting documents, generate plausible professional histories and online profiles, and produce consistent writing styles across communications.

Remote Hiring Fraud

One of the most significant emerging threats for enterprises, in my view, is fraud in remote hiring and onboarding. U.S. government agencies have publicly described schemes in which operatives working on behalf of the North Korean government obtained remote IT jobs at Western companies using stolen or fabricated identities, sometimes with AI-altered photographs and, by some accounts, AI assistance

during interviews.⁷ Once hired, these individuals receive legitimate corporate identities, devices, and access. They pass through the organization's identity controls not by defeating them but by being enrolled into them.

This threat highlights a gap that many identity programs overlook: the moment of identity proofing at onboarding. Most enterprise identity controls assume that the person enrolled as an employee is who they claim to be. If that assumption is wrong, every subsequent control, including phishing-resistant authentication, protects the impostor's access rather than preventing it. The same concern applies to contractors and vendors, who are often onboarded with less scrutiny than employees.

Attacks on Identity Verification

Remote identity verification services, which check a government identity document and compare it to a live selfie or video, have become common for customer onboarding, workforce onboarding, and account recovery. These services face growing pressure from AI. Attackers may present AI-generated or manipulated document images,

7. U.S. Department of Justice, "Justice Department Announces Coordinated, Nationwide Actions to Combat North Korean Remote Information Technology Workers' Illicit Revenue Generation Schemes," press release, Office of Public Affairs, June 30, 2025, <<https://www.justice.gov/opa/pr/justice-department-announce-s-coordinated-nationwide-actions-combat-north-korean-remote>>, accessed October 22, 2026.

use deepfake video in place of a live camera feed, or inject synthetic media directly into the video stream so that the verification system never receives genuine camera input. Vendors have developed liveness detection and injection detection techniques in response, and standards bodies have begun to define testing requirements for these capabilities.⁸ In my experience, the quality of these defenses varies considerably across providers, and organizations should evaluate them with the same rigor they apply to other security controls rather than treating identity verification as a commodity service.

BEST PRACTICES THAT WORK

Treat workforce identity proofing as a security control owned jointly by HR, IT, and security, rather than as an HR formality. In practice, this tends to mean verifying identity documents through a provider with demonstrated resistance to deepfake and injection attacks; requiring at least one in-person or supervised verification step for roles with significant access where feasible; checking that the person who interviewed is the person who is onboarded and who receives equipment; scrutinizing shipping addresses for corporate devices; watching for signs such as reluctance to appear on camera, inconsistent backgrounds, or remote access tools installed on new devices; and applying comparable scrutiny to contractors and vendor personnel. These measures add some friction to hiring, but they protect the foundation on which every other identity control rests.

8. European Committee for Standardization, *Biometric Data Injection Attack Detection*, CEN/TS 18099:2024, CEN, Brussels, December 2024.

Attacks Against Identity Infrastructure, Authentication Workflows, and Recovery Processes

As organizations strengthen authentication, attackers increasingly target the systems and processes around it. AI is likely to make several of these attacks more effective.

Identity Infrastructure

The identity provider, directory, federation configuration, and synchronization services described in Chapter 3 remain the highest-value targets in most environments. AI can help attackers analyze complex configurations, identify misconfigurations and escalation paths, and generate tooling more quickly. AI-assisted vulnerability discovery may also shorten the time between the disclosure of a vulnerability in identity software and its exploitation. These developments reinforce the importance of treating identity infrastructure as tier-zero, applying patches quickly, monitoring configuration changes, and reducing the number of people and identities with administrative control.

Authentication Workflows

Attackers look for the weakest path an authentication system will accept. AI can help them find it, by systematically testing different sign-in flows, legacy protocols, fallback methods, and device registration processes. Adversary-in-the-middle phishing kits continue to evolve, and AI can help attackers build convincing lookalike pages tailored to a specific organization's sign-in experience. OAuth consent phishing and device code phishing, described in Chapter 4, can

be made more persuasive with AI-generated pretexts. The defensive principle remains the same: enforce phishing-resistant methods consistently, and remove or restrict every weaker path that could be used to bypass them.

Recovery Processes

Account recovery, as Chapter 4 emphasized, is frequently the weakest point in authentication. AI makes it weaker still if recovery depends on human judgment of a voice, a face, or a plausible story. Attackers may use cloned voices to call help desks, deepfake video to pass remote verification for credential resets, and AI-generated documents to support claims. Organizations should review recovery processes with the assumption that an attacker can convincingly imitate the legitimate user's voice and appearance, and should ensure that recovery for sensitive accounts requires possession of something the attacker is unlikely to have, such as a second enrolled authenticator, or confirmation by a trusted party through an independent channel.

Attacks Through AI Systems

Finally, the AI systems that organizations deploy create new paths to identity compromise. Chapter 9 discussed prompt injection against AI agents. Attackers may also target AI assistants used by help desks and IT staff, attempting to manipulate them into revealing information or taking actions; compromise the credentials and API keys used to access AI services; or poison the data sources that AI systems rely on. As AI becomes embedded in identity processes themselves, including access reviews, threat detection, and support, these systems become

part of the identity attack surface. Chapter 11 returns to this topic from the defender's perspective.

REALISTIC THREAT SCENARIO

An attacker targets a regional bank's IT operations team. Using public information and a previous breach of a third-party service, the attacker compiles details about a systems administrator, including her manager's name, employee ID format, and recent project work. From a conference presentation she gave, the attacker obtains enough audio to clone her voice.

The attacker calls the IT help desk late on a Friday, using the cloned voice and claiming that her phone was lost while traveling. The help desk agent asks for her employee ID and manager's name, which the attacker provides, and then, following procedure, sends a verification link to her personal email address on file. The attacker has already compromised that personal account using a password from the earlier breach. The agent resets her MFA registration.

The attacker enrolls a new authenticator, signs in, and uses her administrative access to create a new privileged account and register a new OAuth application with broad directory permissions. An AI tool helps the attacker rapidly map the bank's cloud environment and identify systems containing customer financial data.

The bank's identity threat detection flags the sequence of an MFA reset followed by new privileged account creation and OAuth application registration within an hour, and the security team contains the incident before data is exfiltrated. A review concludes that the recovery procedure relied on knowledge an attacker could obtain and on a channel the attacker could

control. The bank changes its procedure for administrators to require in-person verification or confirmation through a video call with the manager initiated from a known corporate number, combined with possession of a backup hardware key issued at onboarding.

Threat Modeling for AI-Era Attacks and Prioritizing Defensive Controls

Faced with this range of threats, security leaders need a disciplined way to decide where to invest. Threat modeling provides one.

Revisiting Trust Assumptions

I find it useful to begin by listing the trust assumptions embedded in the organization's identity processes, and then asking whether each one still holds. Do we assume that a familiar voice on the phone is the person it sounds like? That a person on a video call is real? That a well-written email from a colleague is genuine? That a person who passed a remote document check is who they claim to be? That knowledge of personal details demonstrates identity? That a legitimate sign-in from a known user represents that user's intent? Each assumption that no longer holds reliably points to a process that may need to change.

Mapping Attacks to Processes

Rather than modeling threats against technology alone, I recommend modeling them against the business and identity processes that at-

tackers are likely to target: onboarding, authentication, account recovery, privilege elevation, payment and banking changes, executive requests, vendor onboarding, and the deployment of AI agents. For each process, consider how an attacker equipped with AI might defeat it, what the impact would be, and which controls would prevent, detect, or limit the attack. Frameworks such as MITRE ATT&CK, and its companion MITRE ATLAS for threats to AI systems, can help structure this analysis.⁹

Prioritizing Controls

In my experience, a relatively small set of controls addresses a large share of AI-era identity risk, and I would generally prioritize them as follows.

First, phishing-resistant authentication for privileged users and other high-value identities, with weaker fallback methods removed. This directly counters AI-enhanced phishing and credential relay.

Second, hardened account recovery and help desk procedures that do not depend on knowledge, voice, or appearance alone. This counters voice cloning and deepfake impersonation at the point where it is most dangerous.

Third, out-of-band verification for high-impact business actions, such as payment changes, large transfers, and urgent executive requests, through channels established in advance. This counters im-

9. The MITRE Corporation, *MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems*, <<https://atlas.mitre.org/>>, accessed October 22, 2026.

personation-based fraud regardless of how convincing the impersonation is.

Fourth, stronger identity proofing at onboarding for employees, contractors, and vendors, using verification services evaluated for resistance to synthetic media. This counters synthetic identities and hiring fraud.

Fifth, identity threat detection tuned for rapid post-compromise activity, including MFA changes, new privileged accounts, new OAuth applications, and unusual data access. This shortens the window that AI-accelerated attackers can exploit.

Sixth, least privilege and just-in-time access, which limit what any successfully compromised identity can do, whether human, machine, or agent.

I would also encourage organizations to update security awareness programs. Teaching people to spot imperfections in fakes is becoming less reliable. Teaching them to follow verification procedures for sensitive requests, regardless of how convincing the request appears, and to feel empowered to slow down when urgency is applied, is likely to remain effective. The treasury analyst in the story that opened this chapter did not detect the deepfake. She followed a process, and the process worked.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Detect the fake, or make the fake irrelevant? Deepfake and synthetic media detection tools can help identify manipulated audio, video, and documents, and they may be valuable in some contexts, such as remote identity verification. However, detection is an arms race, and its accuracy may degrade as generation techniques improve.

Process and cryptographic controls take a different approach. Rather than trying to determine whether a voice or face is genuine, they make the decision depend on something an impostor is unlikely to possess: an enrolled hardware authenticator, a callback to a number registered in advance, approval from a second person through an independent system, or a signed request.

In my view, organizations should treat detection as a supplementary layer and invest primarily in controls that make impersonation irrelevant to the outcome. A fraud that depends on persuading someone to bypass a cryptographic or multi-party control is far harder to execute than one that depends only on sounding convincing.

PRACTITIONER CHECKLIST

Which identity and business processes rely on recognizing a person's voice, face, or writing style, or on knowledge of personal information, as the primary means of verification?

How does your help desk verify identity before resetting credentials or MFA for privileged users, and could a cloned voice and leaked personal data defeat it?

Are high-impact business actions, such as payment and banking changes, protected by out-of-band verification through channels established in advance?

How are employees, contractors, and vendor personnel verified at onboarding, and has your identity verification provider been evaluated for resistance to deepfake and injection attacks?

Does your identity threat detection alert quickly on sequences such as an MFA reset followed by privilege changes or new OAuth applications?

Have you updated security awareness training to emphasize process over detection of fakes?

Have you threat-modeled your identity processes against attackers equipped with AI, and do you have a prioritized plan for the resulting gaps?

Looking Ahead

This chapter has focused on how attackers are using AI against identity. It would be incomplete, and somewhat discouraging, to stop there. Defenders also have access to AI, and it offers real opportunities to see more, act faster, and manage identity at a scale that manual processes cannot reach.

Chapter 11 turns to those opportunities. It examines how AI and machine learning can strengthen identity analytics and anomaly detection, uncover excessive privileges and dormant accounts, automate governance and remediation workflows, and integrate with identity threat detection, SIEM, and security operations. It also addresses the risks of AI-assisted security, including false positives, unreliable recommendations, and adversarial manipulation, and how to apply AI in ways that strengthen rather than undermine human judgment.

11. Using AI to Strengthen Identity Security

Several years ago, I worked with an identity team that was responsible for quarterly access reviews covering tens of thousands of employees and several million individual entitlements. The team was small and capable, and it was losing. Each review cycle consumed weeks of effort from the team and from thousands of managers, and at the end of it, the revocation rate was barely above zero. When we analyzed the results, we found that the managers who revoked the most access were not the most diligent ones. They were the ones who happened to manage small teams with simple roles. The managers responsible for complex, high-risk access had approved nearly everything, almost certainly because they had no practical way to evaluate it.

We introduced a relatively simple analytical approach. For each entitlement, we compared it with the access held by the person's peers in the same role and department, checked whether it had been used recently, and flagged combinations that violated separation of duties

rules. Instead of asking managers to review everything, we asked them to review the small fraction of access that was unusual, unused, or risky, and we showed them why each item had been flagged. Review effort fell substantially, and revocations rose sharply. Nothing about the technology was especially advanced. What changed was that we used data and analytics to direct limited human attention toward the decisions that mattered.

I begin with that story because I think it captures both the promise and the right framing of AI in identity security. The previous chapter described how attackers are using AI against identity. Defenders have the same tools available, and in some respects they have advantages: they own the data, they understand the environment, and they can embed AI into processes at scale. Used well, AI can help security teams see more, decide better, and act faster. Used poorly, it can produce confident errors, automate bad decisions, and introduce new risks.

This chapter examines how AI and machine learning can strengthen identity analytics and anomaly detection, help discover excessive privileges, dormant accounts, and hidden identity relationships, automate governance, access reviews, and remediation, and integrate with identity threat detection, SIEM, and security operations. It closes with the risks of AI-assisted security and how to manage them.

AI and Machine Learning for Identity Analytics and Anomaly Detection

Identity generates enormous volumes of data: authentication events, token issuance, access requests, permission changes, application activity, and governance decisions. Most of that data is never examined by a person. Machine learning has been applied to identity analytics for

years, well before the current wave of interest in generative AI, and it remains the foundation for many of the most useful capabilities.

Behavioral Baselines

Chapter 7 introduced behavioral detection. Machine learning makes it practical at scale by building baselines for each identity, or for groups of similar identities, across many dimensions at once: typical sign-in times and locations, devices, applications, data volumes, administrative actions, and sequences of activity. Deviations from these baselines can indicate a compromised account, an insider acting outside their normal scope, or a non-human identity being misused.

Peer group analysis is particularly useful. Rather than comparing an identity only with its own history, which may be short or already compromised, models compare it with others who share the same role, department, or function. An accountant who suddenly accesses engineering source code stands out against other accountants, even if it is the first time either has done anything at all.

Risk Scoring

Many identity platforms combine multiple signals into risk scores for sign-ins, sessions, and identities. A sign-in from an unfamiliar device, at an unusual hour, from infrastructure associated with anonymizing services, shortly after the user's credentials appeared in a leak, might receive a high score that triggers step-up authentication or blocks access. Identity-level risk scores can incorporate posture factors such as privilege level, MFA strength, and dormant entitlements alongside behavioral signals.

Risk scoring works best, in my experience, when it drives proportionate responses rather than binary decisions, and when the factors contributing to a score are visible to the people acting on it. An analyst who sees only a number is less able to judge whether it is meaningful than one who sees the specific signals behind it.

Where Generative AI Adds Value

Generative AI and large language models contribute something different from traditional machine learning. They are less suited to detecting statistical anomalies in high-volume telemetry and more suited to interpretation: explaining what a sequence of events means, summarizing an identity's activity in plain language, translating cryptic entitlement names into business descriptions, and allowing analysts to query identity data in natural language rather than specialized query syntax. In my view, the most effective architectures combine both: traditional models to detect and score, and language models to explain, investigate, and assist.

ENTERPRISE FIELD NOTES

When organizations evaluate AI-based identity analytics, I encourage them to ask vendors and internal teams a simple question about each detection: what would an analyst do with this alert, and what information would they need to do it? Detections that arrive with clear context, including the specific behaviors that triggered them, the identity's privilege level, and recent related changes, tend to be acted on. Detections that arrive as unexplained scores tend to be ignored after the first few false positives. The value of an analytic is determined less by its sophistication than by whether it leads to a good decision.

Discovering Excessive Privileges, Dormant Accounts, and Hidden Identity Relationships

Earlier chapters emphasized how difficult it is to understand the true state of access in a large enterprise. This is one of the areas where analytics and AI offer the clearest and most immediate value, because the problem is fundamentally one of finding patterns in very large, complex data sets.

Excessive and Unused Privileges

By correlating granted permissions with actual usage, analytics can identify access that has not been exercised over a defined period and is therefore a candidate for removal. In cloud environments, where identities may hold thousands of permissions, this analysis is effectively impossible to perform manually. Peer comparison adds another dimension, highlighting access that is unusual for a person's role even if it is used, which may indicate accumulated access from previous roles or inappropriate grants.

Dormant and Orphaned Accounts

Dormant accounts, which have not authenticated for an extended period, and orphaned accounts, which have no valid owner, are attractive targets for attackers and straightforward to detect with the right data. Correlating directory and application accounts against authoritative sources such as HR records and the non-human identity inventory described in Chapter 8 can reveal accounts belonging to departed employees, expired contractors, decommissioned applications, and

identities whose owners have left. Machine learning can help match accounts to owners when naming is inconsistent, a common problem when accounts were created across many systems over many years.

Hidden Relationships and Attack Paths

Some of the most important insights come from analyzing identities as a graph. Identities, groups, roles, permissions, resources, and systems are nodes, and the relationships among them are edges: membership, ownership, delegation, the ability to reset a password, the ability to assume a role, or the ability to read a secret that grants other access. Graph analysis can reveal paths that allow a seemingly unprivileged identity to reach tier-zero systems through a chain of indirect relationships, and it can identify the small number of nodes, such as a single overly privileged group or service account, whose removal would break many such paths at once.

Attackers have used graph-based tools to map directory escalation paths for years. Defenders increasingly use similar techniques, extended across on-premises directories, cloud platforms, and SaaS applications.¹ AI can help prioritize findings, explain paths in plain language, and recommend which relationships to change for the greatest reduction in risk.

1. SpecterOps, Inc., "Introduction," *BloodHound Documentation*, <<https://bloodhound.specterops.io/get-started/introduction>>, accessed October 22, 2026.

Role Mining

Chapter 5 described the difficulty of defining roles in large organizations. Role mining applies clustering and pattern analysis to existing access data to propose candidate roles based on what groups of people actually hold. Role mining has been available for many years and has a mixed record, largely because existing access often reflects accumulated excess rather than genuine need, so mined roles can encode the very problems they are meant to fix. Combining role mining with usage data and peer analysis, and treating its outputs as proposals for business review rather than final definitions, tends to produce better results.

Automating Identity Governance, Access Reviews, and Remediation

Smarter Access Reviews

As the opening story illustrated, AI can transform access reviews from exhaustive exercises into focused ones. Analytics can pre-approve low-risk, recently used access that is consistent with peers, flag unusual or unused access for attention, and provide reviewers with context explaining each recommendation. Language models can translate technical entitlement names into descriptions a business manager can understand. The reviewer remains the decision-maker, but their attention is directed where it is most valuable.

There is a risk worth naming. If reviewers come to accept AI recommendations without reflection, the review becomes a rubber

stamp of the model's judgment rather than the manager's, and errors in the model propagate unchecked. Designs that require reviewers to make an active decision on flagged items, that sample some pre-approved items for human verification, and that track how often reviewers override recommendations help guard against this.

Access Requests and Provisioning

AI can also improve access requests. When a user requests access, a model can recommend the appropriate entitlement based on what peers in similar roles hold, warn when a request would create a separation of duties conflict, suggest time-limited rather than permanent access, and route high-risk requests for additional approval. Natural language interfaces allow users to describe what they need to do rather than navigating catalogs of technical entitlements. These capabilities can reduce both over-provisioning and the frustration that drives users to request overly broad access.

Remediation

Automated remediation is where AI-assisted identity security can deliver significant gains and also where the risks are greatest. Removing unused permissions, disabling dormant accounts, rotating exposed secrets, and revoking risky OAuth consents are all actions that can be automated. They are also actions that can disrupt business operations if performed incorrectly.

In my experience, the most successful approaches use graduated automation. At the lowest level, the system recommends actions and humans perform them. At the next level, the system prepares actions and humans approve them. At a higher level, the system performs

actions automatically within defined boundaries, such as disabling accounts that have been dormant for an extended period and have no owner, with notification and easy reversal. The most consequential actions, such as changes to privileged access or production service accounts, remain under human control. Organizations can move individual use cases up this ladder as confidence grows, based on measured accuracy and the impact of errors.

BEST PRACTICES THAT WORK

Before automating any identity remediation, define three things explicitly: the precise conditions under which the action will be taken, the mechanism for reversing it quickly, and the measurement that will tell you whether the automation is making correct decisions. Run the automation in recommendation-only mode for a period and compare its recommendations with what experienced practitioners would have done. Begin automatic execution with the lowest-risk, most reversible actions, and expand gradually. Record every automated action with the evidence and logic that produced it, so that both auditors and incident responders can understand what happened and why.

Integrating AI With ITDR, SIEM, and Security Operations

Security operations centers face a persistent imbalance between the volume of alerts and the number of analysts available to investigate them. Identity-related alerts add to that volume, and as Chapter 7 discussed, they often require specialized knowledge to interpret. AI offers several ways to help.

Correlation and Triage

Machine learning can correlate identity signals with endpoint, network, cloud, and application telemetry to assemble related events into a single incident, rather than presenting analysts with dozens of disconnected alerts. A risky sign-in, a new MFA registration, a mailbox rule, and an unusual file download, each of which might be a low-priority alert on its own, can be recognized together as a probable account takeover. AI can also help prioritize incidents based on the privilege of the identities involved, the sensitivity of the resources accessed, and the presence of known attack patterns.

AI Assistants for Analysts

Generative AI assistants embedded in security platforms can summarize incidents, explain unfamiliar events, reconstruct an identity's activity timeline across systems, draft queries, suggest next investigative steps, and prepare incident reports. For identity investigations, which frequently require understanding of directory structures, token flows, and cloud permission models, these assistants can help less experienced analysts work effectively and allow experienced analysts to work faster. In my experience, the gains are most pronounced in the time-consuming tasks of gathering and summarizing evidence.

Agentic Security Operations

The industry is moving toward security agents that can carry out multi-step investigations and responses with limited human direction: gathering evidence, forming hypotheses, querying additional

data sources, and, within defined limits, taking containment actions such as revoking sessions or disabling accounts. These capabilities may help defenders match the speed of AI-accelerated attackers described in Chapter 10.

They also bring every concern from Chapter 9 into the security operations center. A security agent that can disable accounts, revoke tokens, and change configurations is a highly privileged non-human identity. It processes untrusted content, including attacker-controlled data in logs, emails, and files, and may therefore be susceptible to prompt injection. An attacker who could manipulate a security agent might cause it to ignore malicious activity, disrupt legitimate users, or take destructive actions. Security agents should be governed with at least the same rigor as any other privileged agent: their own identities, narrowly scoped permissions, approval requirements for high-impact actions, comprehensive logging, and the ability to be halted quickly.

ARCHITECTURE DECISIONS AND TRADE-OFFS

How much autonomy should AI have in identity response? Fully autonomous response can contain attacks within seconds, which may be decisive against fast-moving, automated adversaries. It also risks disrupting legitimate users, executives, and business-critical service accounts on the basis of false positives, and it creates a powerful target for manipulation.

Human-approved response preserves judgment and accountability but may be too slow for some attacks, particularly outside business hours.

A balanced approach defines response tiers. Low-impact, reversible actions, such as forcing re-authentication or revoking a single user's sessions, can be automated for high-confidence detections. Moderate actions, such as disabling a standard user

account, can be automated with immediate notification and an easy path to reversal. High-impact actions, such as disabling privileged or service accounts, rotating federation certificates, or acting on executives, should require human approval. The tiers should be reviewed regularly based on measured accuracy and business impact.

Managing the Risks of AI-Assisted Security

AI is not a neutral addition to the security stack. It introduces its own failure modes, and organizations that adopt it without understanding them may end up less secure rather than more.

False Positives and False Negatives

All detection models make errors. False positives consume analyst time, frustrate users, and, over time, train people to ignore alerts. False negatives allow attacks to proceed while creating a false sense of coverage. Organizations should measure both, using known incidents, red team exercises, and periodic review of closed alerts, rather than relying on vendor claims. Models also drift as the environment changes, so performance measured at deployment may not persist.

Unreliable Recommendations

Language models can produce explanations and recommendations that sound authoritative but are wrong. An assistant may misinterpret an event, fabricate a detail, or recommend an action that is inappropriate for the specific environment. In identity security, where

an incorrect recommendation might lead to removing critical access or overlooking a compromise, this is a serious concern. AI outputs should be grounded in the organization's actual data, should cite the evidence behind them, and should be treated as input to human judgment rather than as conclusions, particularly for consequential decisions.

Adversarial Manipulation

Attackers who know that defenders use AI will attempt to evade or exploit it. They may operate slowly and within normal patterns to avoid behavioral detection, gradually shift baselines so that malicious activity appears normal, or plant content designed to manipulate AI assistants and agents that process it. Defenses include combining AI-based detection with deterministic rules for high-risk events, so that critical changes such as new federation trusts always generate alerts regardless of model output; protecting the integrity of training and baseline data; and treating content processed by security AI as untrusted.

Data Exposure

Identity and security data is among the most sensitive information an organization holds. Sending it to AI services raises questions about where it is processed, whether it is retained or used for training, who can access it, and whether its use complies with privacy and data residency requirements. Organizations should understand the data handling terms of any AI service used in security operations and should apply the same access controls to AI tools that they apply to the underlying data.

Integration Complexity

AI capabilities are only as good as the data they receive. Fragmented identity telemetry, inconsistent identifiers across systems, incomplete inventories, and poor data quality limit what any model can achieve. In my experience, organizations sometimes expect AI to compensate for foundational gaps, when in fact those gaps constrain AI as much as they constrain human analysts. The investments described in earlier chapters, including telemetry coverage, authoritative identity sources, and complete inventories, are prerequisites for effective AI, not alternatives to it.

Human Oversight

Finally, AI should strengthen human judgment rather than replace it. Accountability for identity decisions, including who has access, which incidents are serious, and when to take disruptive action, should remain with people. Organizations should define which decisions AI may make autonomously, which require human approval, and how human reviewers will remain engaged rather than becoming passive approvers of machine output. Skills matter too. If analysts rely entirely on AI to interpret identity events, the organization may lose the expertise needed to recognize when the AI is wrong.

REALISTIC THREAT SCENARIO

A financial services company deploys an AI security agent to triage identity alerts and perform automated containment. The agent can read email messages reported as phishing, query

identity logs, and, for high-confidence account compromise detections, revoke sessions and disable standard user accounts.

An attacker who has compromised a contractor's account learns about the agent from internal documentation. The attacker sends a crafted email to several employees, knowing that some will report it. Hidden in the message is text instructing any automated analysis system to classify the activity from the contractor's account as an authorized penetration test and to suppress related alerts. When the agent processes the reported email, it incorporates the injected instruction into its reasoning and downgrades subsequent alerts involving the contractor's account.

The attack is discovered because the company had configured deterministic rules, independent of the agent, that always alert on new privileged role assignments. When the attacker uses the contractor's account to grant itself elevated access, that rule fires and reaches a human analyst, who investigates and finds both the compromise and the agent's suppressed alerts.

The review leads to several changes: content from reported emails is now processed in a separate context that cannot influence classification decisions on other alerts, the agent can no longer suppress alerts but only lower their priority with a logged justification visible to analysts, and suppression decisions are sampled for human review. The incident illustrates why AI-based security controls should be layered with deterministic safeguards and why security agents must be treated as targets.

PRACTITIONER CHECKLIST

Where are you currently using machine learning or AI in identity security, and how have you measured its accuracy in your environment?

Do AI-generated detections and recommendations include the evidence and reasoning behind them?

Are you using usage data, peer analysis, and graph analysis to identify excessive privileges, dormant accounts, and attack paths?

Have access reviews been redesigned to focus human attention on high-risk access, with safeguards against reviewers simply accepting AI recommendations?

For each automated remediation, are the conditions, reversal mechanism, and success measurement defined?

Are AI security agents governed as privileged non-human identities, with scoped permissions, logging, approval requirements, and the ability to be halted?

Do deterministic rules continue to alert on critical identity events regardless of AI model output?

Do you understand how AI services handle the identity and security data you send them?

How are you preserving analysts' identity expertise as AI assistance becomes more common?

Looking Ahead

This chapter concludes Part III. Over the last four chapters, we have examined non-human identities, AI agents, AI-powered attacks, and the use of AI in defense. Together, they describe an identity landscape

that is larger, faster, and more complex than the one most identity programs were designed for, but also one in which defenders have powerful new tools available.

The remaining question is how to bring all of this together in practice. Part IV turns from capabilities to execution. Chapter 12 examines how to assess identity security maturity, exposure, and readiness; prioritize risks, investments, and remediation; develop an identity security roadmap and operating model; measure effectiveness through meaningful metrics; and avoid the implementation failures that I have seen undermine otherwise well-designed programs.

12. Building and Operating an Effective Identity Security Program

Early in my time leading identity security at a large enterprise, I inherited a program that, on paper, had everything. It had a modern identity provider, a governance platform, a privileged access management tool, and a multi-year roadmap that had been presented to the board. It also had a growing backlog of unfinished integrations, a governance platform connected to a fraction of the organization's applications, a PAM deployment covering a small share of privileged accounts, and a team that was exhausted from running several large projects at once while also handling daily operations. The roadmap described an end state that everyone agreed was desirable. It did not describe how to get there from where the organization actually stood, which risks to address first, or who outside the identity team needed to do what.

We spent the next several months doing something that felt, at the time, like going backward. We assessed where we really were. We identified the handful of exposures that represented most of our risk. We stopped or slowed several projects that were consuming effort without reducing much exposure. We clarified who owned which decisions across security, IT, application teams, and the business. And we chose a small number of metrics that would tell us, honestly, whether we were making progress. The program that emerged was less ambitious on paper and far more effective in practice.

The previous eleven chapters described capabilities: architecture, authentication, authorization, privileged access, detection and response, non-human identities, AI agents, emerging threats, and AI-assisted defense. This chapter addresses the question that determines whether any of those capabilities make a difference: how to build and operate an identity security program that improves steadily over time. It examines how to assess maturity, exposure, and readiness; how to prioritize risks, investments, and remediation; how to develop a roadmap and operating model; how to measure effectiveness; and how to avoid the implementation failures I have seen most often.

Assessing Identity Security Maturity, Exposure, and Readiness

Every effective program begins with an honest understanding of its starting point. I find it useful to assess three distinct dimensions, because each answers a different question and organizations frequently confuse them.

Maturity

Maturity describes how well established the organization's identity capabilities and processes are. Are capabilities ad hoc, defined, consistently implemented, measured, or continuously optimized? Several frameworks can help structure this assessment. The CISA Zero Trust Maturity Model includes an identity pillar with defined stages from traditional to optimal.¹ The NIST Cybersecurity Framework, particularly in its 2.0 revision with its added emphasis on governance, provides a broader structure into which identity capabilities fit.² Industry groups and vendors offer identity-specific maturity models as well. Any of these can be useful, provided the organization treats the result as a guide rather than a score to be optimized.

A word of caution: maturity assessments tend to overstate capability when they rely on self-reporting and on the existence of policies or tools rather than their actual coverage and effectiveness. An organization may have a PAM platform, a defined process for privileged access, and a policy requiring its use, and still have the majority of its privileged accounts outside the platform. I generally recommend that

-
1. Cybersecurity and Infrastructure Security Agency, *Zero Trust Maturity Model*, Version 2.0, CISA, Washington, D.C., April 2023, <https://www.cisa.gov/sites/default/files/2023-04/zero_trust_maturity_model_v2_508.pdf>, accessed October 22, 2026.
 2. National Institute of Standards and Technology, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST Cybersecurity White Paper 29, NIST, Gaithersburg, Md., February 26, 2024, <<https://doi.org/10.6028/NIST.CSWP.29>>, accessed October 22, 2026.

maturity assessments include evidence of coverage and effectiveness, not only of design.

Exposure

Exposure describes the organization's actual identity risk as it exists today. Where maturity asks how capable the organization is, exposure asks where an attacker would succeed. Useful exposure indicators include the proportion of users and privileged users without phishing-resistant authentication; the number of accounts exempt from MFA; the volume of standing privileged access; the number of dormant, orphaned, and departed-user accounts; the number of non-human identities with long-lived credentials and no owner; exposed secrets in code and shared locations; attack paths from ordinary identities to tier-zero systems; unreviewed third-party integrations; and AI agents operating with broad access and no registry entry.

Exposure can often be measured directly from data, which makes it more objective than maturity. In my experience, an exposure assessment frequently produces the most persuasive material a security leader can bring to executives, because it describes concrete, specific weaknesses rather than abstract capability levels.

Readiness

Readiness describes the organization's capacity to change. Does the identity team have the skills and staffing to execute its plans? Are application owners willing and able to integrate their systems? Does the HR system provide reliable data? Is there executive sponsorship? Can the help desk absorb the impact of an authentication rollout? Are there competing initiatives, such as a cloud migration or an ERP

replacement, that will consume the same people? Readiness often determines the pace at which a program can realistically move, and plans that ignore it tend to slip repeatedly.

ENTERPRISE FIELD NOTES

One of the most effective assessment techniques I have used is also one of the simplest. I ask the team to trace a small number of specific scenarios end to end with evidence rather than assertions: a phished user with administrative rights, a departing contractor, a leaked cloud key, a new AI integration approved through an OAuth consent screen, and a help desk call requesting an MFA reset for an executive. For each, we document what would actually happen, which controls would apply, and where the gaps are. A handful of these traces often reveals more about the real state of a program than a lengthy questionnaire, and the results are concrete enough that executives and application owners understand them immediately.

Prioritizing Risks, Investments, and Remediation

No organization can address every identity weakness at once. Prioritization is, in my view, the single most important skill in running an identity security program, and the one most often neglected in favor of following a vendor's or framework's sequence.

A Risk-Based Approach

I recommend prioritizing based on a combination of three factors: the likelihood that a weakness will be exploited, the impact if it is, and the

effort required to address it. Likelihood draws on threat intelligence and the attack patterns described in Chapters 2 and 10. Weaknesses that attackers exploit routinely, such as missing MFA, phishable authentication for administrators, and weak help desk verification, deserve higher priority than theoretical concerns. Impact draws on the business mapping described at the end of Chapter 2: weaknesses that expose critical processes, sensitive data, or tier-zero systems matter more than those affecting low-value resources. Effort reflects cost, complexity, disruption, and dependency on other teams.

Common High-Priority Items

Although every environment differs, I find that a consistent set of improvements appears near the top of most prioritized lists. These include protecting identity infrastructure as tier-zero and limiting its administrators; enforcing phishing-resistant authentication for privileged users and removing weaker fallbacks; hardening account recovery and help desk procedures; eliminating MFA exemptions or reducing them to documented, monitored exceptions; removing standing privileged access in favor of just-in-time elevation; disabling dormant and departed-user accounts; finding and rotating exposed secrets, starting with those that grant privileged access; establishing identity telemetry and detection for critical identity changes; and inventorying third-party integrations and AI agents with access to sensitive data. Many of these are relatively inexpensive compared with large platform projects, and they often reduce a disproportionate share of exposure.

Balancing Quick Wins and Foundational Work

Prioritization must balance quick, visible reductions in risk against the foundational work that enables future progress, such as improving HR data quality, building authoritative inventories, and expanding governance connectivity. Programs that pursue only quick wins eventually stall because the foundations cannot support further improvement. Programs that pursue only foundations struggle to demonstrate value and lose sponsorship. In my experience, a healthy portfolio includes both, with each foundational investment tied explicitly to the risk reductions it will enable.

ARCHITECTURE DECISIONS AND TRADE-OFFS

Platform consolidation or targeted risk reduction? Large platform programs, such as replacing a governance tool or consolidating identity providers, can deliver lasting structural benefits. They also consume substantial budget, staff time, and organizational attention for long periods, during which existing exposures may remain unaddressed.

Targeted risk reduction, such as removing MFA exemptions, enforcing phishing-resistant authentication for administrators, or cleaning up dormant accounts, delivers measurable improvement quickly but may be constrained by existing platform limitations.

In my experience, organizations do best when platform decisions are driven by the risk reductions they enable rather than pursued for their own sake. Before committing to a major platform change, ask which specific exposures it will reduce, by how

much, and when, and whether the same reductions could be achieved sooner with the tools already in place.

Developing an Identity Security Roadmap and Operating Model

Structuring the Roadmap

A useful identity security roadmap translates prioritized risks into sequenced initiatives with clear outcomes, owners, dependencies, and time frames. I generally recommend organizing it into three horizons.

The first horizon, roughly the first six months, focuses on reducing the most acute exposures with existing tools and establishing the visibility needed for later phases. Typical initiatives include hardening identity infrastructure, enforcing phishing-resistant authentication for administrators, tightening help desk procedures, removing dormant accounts, scanning for and rotating exposed secrets, and establishing core identity detections.

The second horizon, roughly six to eighteen months, extends controls across broader populations and builds foundations. Typical initiatives include expanding phishing-resistant authentication to high-risk business roles and then the general workforce, implementing just-in-time access for privileged roles, expanding governance coverage to high-risk applications, improving mover processes, inventorying and assigning ownership to non-human identities, migrating pipeline credentials to workload identity federation, and establishing an AI agent registry and review process.

The third horizon, roughly eighteen to thirty-six months, addresses structural change and optimization. Typical initiatives include consolidating identity platforms where justified, moving toward zero standing privilege, implementing continuous access evaluation, adopting delegated authorization models for AI agents, applying analytics and automation to governance and remediation, and refining the operating model based on experience.

The roadmap should be revisited at least annually, and more often when the threat landscape or the business changes significantly. Given how quickly AI agents and AI-enabled attacks are evolving, I would expect the later horizons in particular to require regular adjustment.

The Operating Model

Technology and roadmaps do not run themselves. An identity security program needs an operating model that defines who is responsible for what, how decisions are made, and how work flows across organizational boundaries.

Identity responsibilities are typically distributed. IAM teams, often within IT, operate directories, identity providers, and governance platforms. Security teams define policy, assess risk, and detect and respond to threats. Infrastructure and cloud teams manage platform-specific identities and permissions. Application owners control fine-grained authorization within their systems. HR owns the authoritative source for workforce identity. Procurement and vendor management influence third-party access. Development teams create many non-human identities and, increasingly, AI agents. Business leaders approve access and accept risk.

In my experience, programs struggle when these responsibilities are unclear, particularly at the boundaries. Who decides whether a

new AI agent may access customer data? Who owns a service account when the application team that created it has been reorganized? Who can approve an exception to the phishing-resistant authentication requirement? A clear operating model answers these questions in advance, usually through a combination of defined ownership, documented decision rights, and governance forums.

Many large organizations have found value in establishing an identity security function or center of excellence that sets standards, owns the roadmap, coordinates across teams, and reports on risk, while operational responsibilities remain distributed. Where that function sits, whether under the CISO, the CIO, or jointly, matters less than whether it has the authority to set standards and the relationships to see them implemented. I do believe that identity security should have a clear line of accountability to the security leadership of the organization, given its central role in enterprise risk.

Finally, the operating model must address people. Identity security requires skills that are in short supply, spanning protocols, directories, cloud IAM, governance, detection, and now AI. Investing in training, cross-training between IAM and security operations staff, and retaining experienced practitioners is, in my experience, as important as any technology investment.

BEST PRACTICES THAT WORK

Make application and platform owners accountable for identity outcomes in their systems, not only the identity team. In practice, this tends to mean publishing clear identity standards for applications, covering federation, MFA, provisioning, logging, privileged access, and non-human identities; requiring new applications and AI agents to meet them before production; reporting each owner's compliance and exposure to their leader-

ship; and providing integration patterns and support that make compliance straightforward. Identity teams cannot integrate and govern thousands of applications alone. Distributed ownership with central standards and visibility scales far better.

Measuring Effectiveness Through Meaningful Metrics

What gets measured tends to get managed, which makes the choice of metrics consequential. In my experience, many identity programs measure activity, such as the number of applications onboarded, reviews completed, or users enrolled, rather than outcomes, such as reduced exposure and faster response. Activity metrics are useful for managing projects, but they can create an illusion of progress while risk remains largely unchanged.

Outcome-Oriented Metrics

I recommend that leadership reporting emphasize a small set of outcome-oriented metrics, supported by operational detail for the teams doing the work. Useful metrics include the following.

For authentication: the percentage of privileged users and of all users protected by phishing-resistant authentication with weaker methods disabled, and the number of MFA exemptions and their age.

For privileged access: the percentage of privileged access that is just-in-time rather than standing, the coverage of privileged accounts by PAM controls, and the number of attack paths from ordinary identities to tier-zero systems.

For lifecycle and governance: the median and maximum time from termination to full access removal across systems, the number of active accounts belonging to departed employees or expired contractors, the volume of unused entitlements, and the number of unresolved separation of duties conflicts.

For non-human identities and agents: the percentage of non-human identities with a current owner, the number of long-lived privileged credentials, the number of exposed secrets discovered and the time to rotate them, and the percentage of AI agents registered and reviewed.

For detection and response: the coverage of critical identity events by detection rules, the time to detect and contain identity incidents, and the results of exercises that test identity response playbooks.

Communicating Metrics

Metrics are most persuasive when they are tied to business risk and presented as trends. A chart showing the reduction in standing privileged access over four quarters, alongside a brief explanation of what that means for the organization's exposure to ransomware, communicates more than a table of raw numbers. As Chapter 2 noted, honesty about uncertainty also matters. Executives and boards tend to trust programs that acknowledge what they do not yet know and explain how they plan to find out.

I would also caution against setting targets that encourage gaming. If the goal is to reduce the number of MFA exemptions, teams may reclassify exemptions rather than eliminate them. If the goal is to complete access reviews on time, reviewers may approve everything. Pairing metrics, for example measuring both review completion and

revocation rates, and periodically validating metrics through independent testing, helps keep them meaningful.

Avoiding Common Implementation Failures and Sustaining Improvement

Over many years and many organizations, I have seen identity programs fail in recurring ways. Recognizing these patterns early is one of the most valuable things a program leader can do.

The first failure is treating identity security as a technology project rather than an ongoing program. Tools are purchased and deployed, the project is closed, and the organization assumes the problem is solved. Identity environments change continuously, and controls that are not actively operated and maintained erode.

The second is pursuing too much at once. Ambitious programs that launch many large initiatives simultaneously tend to overload the identity team and the organization's capacity to absorb change, and they often deliver little until very late, if at all.

The third is neglecting the business. Identity changes affect how people work. Authentication rollouts, access removals, and new request processes that are imposed without communication, training, and attention to user experience generate resistance and exceptions. Programs that involve business stakeholders early and position identity improvements as enablers tend to fare better.

The fourth is building on poor data. Governance, automation, and analytics all depend on accurate identity data. Programs that skip the unglamorous work of cleaning HR feeds, establishing ownership, and building inventories find that their sophisticated tools produce unreliable results.

The fifth is accepting exceptions without limits. Every program needs an exception process, but exceptions that are granted permanently, without owners, compensating controls, or review dates, gradually hollow out the controls they bypass. Many of the attack paths described in this book run through exceptions.

The sixth is focusing on human identities alone. As Chapters 8 and 9 argued, non-human identities and AI agents are a large and growing share of the identity landscape. Programs that address only human users leave much of their exposure untouched.

The seventh is losing momentum. Identity programs that deliver early successes sometimes lose executive attention and funding before structural improvements are complete. Regular reporting on outcomes, linking identity progress to business priorities, and celebrating milestones help sustain support.

Sustaining Continuous Improvement

Sustained improvement depends on feedback loops. Incidents, near misses, red team exercises, audit findings, and metric trends should all feed back into the roadmap. Periodic reassessment of maturity, exposure, and readiness, perhaps annually, helps the program adjust to change. Tabletop exercises that walk through identity-centric scenarios, including compromised administrators, rogue AI agents, and deepfake-enabled help desk attacks, test both controls and decision-making. And a culture that treats identity weaknesses as problems to be solved rather than failures to be hidden encourages the honest reporting that improvement requires.

REALISTIC THREAT SCENARIO

Consider two organizations of similar size facing the same attacker, who uses AI-generated phishing and a cloned voice to target IT administrators.

The first organization has a comprehensive, multi-year identity roadmap and has invested heavily in a new governance platform. Its administrators still use push-based MFA with SMS recovery, it holds substantial standing privileged access, and its help desk verifies callers using employee ID numbers and managers' names. The governance platform is well along in its deployment but not yet connected to the systems the attacker targets. The attacker succeeds in resetting an administrator's MFA through the help desk, uses standing domain administrator rights to deploy ransomware, and the organization spends weeks recovering.

The second organization has a less ambitious roadmap but prioritized differently. In its first six months, it moved administrators to hardware security keys with no SMS fallback, required in-person or manager-verified recovery for privileged accounts, converted most privileged access to just-in-time elevation, and implemented alerts for MFA changes followed by privilege use. The same attack fails at the help desk, and a second attempt using a phishing page fails against the hardware keys. The security team receives an alert about the failed recovery attempt and warns other administrators.

The difference between the two outcomes did not lie in the size of the investment or the sophistication of the platforms. It lay in prioritization: the second organization addressed the exposures attackers were most likely to exploit first.

PRACTITIONER CHECKLIST

Have you assessed maturity, exposure, and readiness separately, using evidence of coverage and effectiveness rather than policy alone?

Is your list of identity risks prioritized by likelihood, impact, and effort, and does it address the exposures attackers most commonly exploit?

Does your roadmap balance quick risk reductions with foundational work, and is each foundational investment tied to specific risk outcomes?

Are ownership and decision rights for identity defined across security, IT, application teams, HR, and the business, including for non-human identities and AI agents?

Are application and platform owners accountable for meeting identity standards in their systems?

Does leadership reporting focus on outcome metrics, presented as trends and tied to business risk?

Are exceptions time-limited, owned, compensated, and reviewed?

Do incidents, exercises, and audit findings feed back into the roadmap on a regular cycle?

Building an effective identity security program is not a single project with a defined end. It is an ongoing discipline that must adapt as the organization, its technology, and its adversaries change. The arrival of AI agents and AI-enabled attacks has made that adaptation more urgent, but the fundamentals of the discipline remain the same: know your identities, protect how they authenticate, limit what they can do,

watch what they actually do, and respond quickly when something goes wrong.

The conclusion that follows steps back from the details of individual capabilities to consider the larger picture. It reflects on why identity will remain foundational to enterprise security, how identity-centric controls must extend to AI agents and other emerging digital actors, how organizations can build security architectures that evolve with new threats, and what the next generation of identity, access, and autonomous computing may require of all of us.

Conclusion: Securing Trust in an Autonomous Enterprise

At the beginning of this book, I described a conference room early in my career where a security team struggled to answer a deceptively simple question about an intrusion: who, or what, was actually acting on the network, and should it have been allowed to do what it did? The attackers in that incident had not broken anything. They had logged in, and every system had done exactly what it was designed to do.

I have thought about that room often while writing these chapters. The technology has changed almost beyond recognition since then. The perimeter that organization relied on has largely dissolved. Its applications would now run in the cloud and in SaaS platforms. Its employees would sign in from anywhere. Its environment would contain far more machine identities than people, and, increasingly, AI agents reading its email, querying its systems, and acting on its behalf.

Yet the question that team could not answer has not changed at all. If anything, it has become the central question of enterprise security.

In this conclusion, I want to step back from the details of individual capabilities and reflect on what I believe the preceding chapters add up to: why identity will remain foundational, how identity-centric controls must extend to AI agents and other emerging actors, how to build an architecture that can adapt, how to turn lessons into sustainable practice, and what the next generation of identity and autonomous computing may ask of us.

Why Identity Will Remain Foundational

It is reasonable to ask whether identity's central role is a passing phase, a product of the cloud transition that will give way to some other organizing principle. I do not believe it is, for a simple reason. Every meaningful security decision ultimately depends on knowing who or what is acting and what it should be permitted to do. Network location once served as a rough proxy for that knowledge. Device posture, behavior, and context now enrich it. But none of these replaces the underlying need to establish an actor's identity and bind it to authority.

That need grows rather than shrinks as computing becomes more distributed and more autonomous. When the number of actors in an environment multiplies, when they operate at machine speed, and when many of them make decisions without a human reviewing each one, the ability to establish trust in each actor, constrain what it can do, and hold someone accountable for its actions becomes more important, not less. Identity is the mechanism through which an organization expresses those decisions.

Throughout this book, I have argued that identity has become both the most important control point in the modern enterprise and one of its most frequently exploited weaknesses. I expect both halves of that statement to remain true for the foreseeable future. Attackers will continue to prefer borrowing trust over defeating it, because borrowing is cheaper and harder to detect. The organizations that fare best will be those that treat identity not as an administrative utility but as a core security discipline, with the investment, expertise, and executive attention that implies.

Extending Identity-Centric Controls to AI Agents and Emerging Digital Actors

The most consequential shift described in this book is the arrival of AI agents as actors within the enterprise. Chapters 8 and 9 argued that agents inherit every weakness in how organizations manage non-human identities and add new ones of their own: non-deterministic behavior, delegated authority on behalf of people, susceptibility to manipulation through the content they process, and the ability to chain actions across systems faster than any human could review them.

It would be easy to conclude that agents require an entirely new security discipline. I do not think that is quite right. What they require is the disciplined extension of principles the identity community has developed over decades, applied with fresh attention to the ways agents differ. Every agent should have its own identity. Every agent should have an accountable owner. Agents should hold the least privilege and the least agency their tasks require, preferably granted just in time and scoped to the person on whose behalf they act. Their actions should be logged in a way that attributes them to both the agent and the delegating human. High-impact actions should require human approval

through processes designed to resist fatigue. And organizations should be able to stop any agent quickly and confidently.

These principles are not new. Least privilege, separation of duties, accountability, and the ability to revoke access are among the oldest ideas in security. What is new is the urgency of applying them consistently to a class of actor that is arriving faster than most governance processes can adapt. In my conversations with security leaders, the organizations that seem best positioned are not those with the most sophisticated AI security tools. They are those that already manage non-human identities well, because the habits of inventory, ownership, scoping, and monitoring transfer directly.

Agents are unlikely to be the last new kind of digital actor. Agents will increasingly interact with other agents across organizational boundaries, negotiating, purchasing, scheduling, and exchanging data on behalf of their principals. Devices and physical systems will act with growing autonomy. Each of these developments will raise the same fundamental questions: who or what is this, on whose authority does it act, what may it do, and who is accountable? Organizations that build identity programs capable of answering those questions for agents today will be better prepared for whatever follows.

Building an Adaptive Security Architecture

If there is one lesson I would draw from the evolution traced in Chapter 1, it is that the architectures we build will be tested by conditions we did not anticipate. The perimeter model was not foolish; it was designed for a world that changed around it. Many identity architectures in use today were designed for a world of human users signing in to applications, and they are now being asked to govern workloads, integrations, and autonomous agents.

An adaptive architecture, in my view, has several characteristics. It is built on open standards, such as OAuth, OpenID Connect, SCIM, FIDO2, and emerging work on workload identity and agent authorization, so that new kinds of actors and applications can be integrated without bespoke mechanisms. It separates policy from enforcement, so that rules can be changed centrally as risks evolve rather than rewritten in every application. It evaluates trust continuously rather than only at sign-in, using signals that can be extended as new sources of risk emerge. It treats identity infrastructure as tier-zero and designs for resilience and recovery, because the more central identity becomes, the more damaging its failure. And it generates rich telemetry by default, so that the organization can see what its identities are doing and detect when something changes.

Perhaps most importantly, an adaptive architecture is designed with the expectation that some controls will fail. No authentication method is beyond circumvention, no detection model is complete, and no agent is immune to manipulation. Layered controls, limited blast radius, rapid detection, and practiced recovery allow an organization to absorb failures without catastrophe. The Realistic Threat Scenarios throughout this book share a common pattern: no single control was decisive, but several overlapping ones would each have reduced the attacker's chance of success. That pattern, I believe, is the essence of a resilient identity architecture.

Turning Lessons Learned Into Sustainable Practice

The recurring features in this book, including the Enterprise Field Notes, Best Practices That Work, and Practitioner Checklists, were intended to bridge the gap between principle and practice. Looking

across them, a handful of lessons stand out as the ones I would most want readers to carry forward.

First, prioritization matters more than ambition. The organizations I have seen make the greatest progress were not those with the most comprehensive roadmaps, but those that identified the exposures attackers were most likely to exploit and addressed them first: phishable authentication for administrators, weak account recovery, standing privilege, dormant accounts, exposed secrets, and unmonitored changes to identity infrastructure.

Second, the weakest path determines the strength of the whole. Strong authentication undermined by a weak recovery process, a governance program that covers only part of the application portfolio, or a PAM deployment that excludes cloud administrators all leave openings that attackers will find. Coverage and consistency deserve as much attention as capability.

Third, data and ownership are the foundations on which everything else rests. Accurate HR feeds, authoritative inventories of human and non-human identities, and clear ownership for every identity and application are unglamorous, but without them, governance, automation, analytics, and response all operate on guesswork.

Fourth, identity security is a team endeavor that extends well beyond the identity team. Application owners, cloud engineers, developers, HR, the help desk, finance, and business leaders all make decisions that shape identity risk. Programs succeed when those groups understand their roles, have clear standards to follow, and are accountable for outcomes.

Fifth, people remain central even as automation grows. AI can direct human attention, accelerate investigation, and automate routine decisions, but accountability for consequential choices should remain with people. The treasury analyst described in Chapter 10,

who stopped a deepfake-enabled fraud by following a verification procedure, is a reminder that well-designed processes and empowered people remain among the most reliable defenses available.

Sustaining these lessons requires treating identity security as a continuous program rather than a series of projects. Environments change, attackers adapt, and controls that are not actively maintained erode. Regular reassessment, honest metrics, feedback from incidents and exercises, and a culture that surfaces weaknesses rather than concealing them are what allow a program to keep improving year after year.

Preparing for the Next Generation of Identity, Access, and Autonomous Computing

I am cautious about predictions. The pace of change in AI over the past few years has surprised many experienced observers, and I expect some of what I have written about agents and AI-enabled attacks to look dated before long. Nonetheless, several developments seem likely enough to merit attention from identity practitioners now.

The first is the continued decline of shared secrets. Passkeys and phishing-resistant authentication for people, and workload identity federation and short-lived credentials for machines, point toward an environment in which passwords and long-lived keys become exceptions rather than the norm. I believe that transition will take many years in large enterprises, but its direction seems clear, and organizations that plan for it deliberately will reduce a large share of their exposure along the way.

The second is the maturation of standards for agent identity and delegation. Work under way in standards bodies and the broader industry on agent authorization, delegation chains, and agent-to-agent

trust is likely to shape how agents are deployed in the coming years. Practitioners who engage with this work, test emerging approaches, and design their architectures to accommodate them will be better positioned than those who wait for finished products.

The third is the growth of verifiable digital credentials. Digital identity wallets and verifiable credentials, which allow individuals and organizations to present cryptographically signed attributes issued by trusted authorities, are advancing through government programs and industry initiatives in several regions.¹ If they achieve broad adoption, they may change how organizations perform identity proofing for employees, customers, and partners, and could offer some defense against the synthetic identities and deepfake-enabled fraud described in Chapter 10.

The fourth is the transition to post-quantum cryptography. Much of identity security depends on public key cryptography: certificates, token signatures, federation assertions, and FIDO credentials. Stan-

-
1. World Wide Web Consortium, *Verifiable Credentials Data Model v2.0*, W3C Recommendation, W3C, May 15, 2025, <<https://www.w3.org/TR/2025/REC-vc-data-model-2.0-20250515/>>, accessed October 22, 2026; *Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 Amending Regulation (EU) No 910/2014 as Regards Establishing the European Digital Identity Framework*, Official Journal of the European Union, L series, April 30, 2024, <<https://eur-lex.europa.eu/eli/reg/2024/1183/oj>>, accessed October 22, 2026.

dards bodies have published post-quantum algorithms,² and governments have set expectations for migration over the coming decade.³ Identity infrastructure, which relies so heavily on signatures and keys, will be an important part of that migration, and organizations should begin by understanding where cryptography is used across their identity systems.

The fifth, and the one I consider most significant, is the deepening of autonomy. As agents take on more consequential tasks, including financial transactions, operational decisions, and interactions with other organizations' agents, the questions of authority, accountability, and trust will become matters not only of security but of governance, law, and business strategy. Identity practitioners will increasingly be asked to help their organizations decide what autonomous systems should be permitted to do, and on whose behalf. I believe that is a role our profession should welcome and prepare for.

2. National Institute of Standards and Technology, "NIST Releases First 3 Finalized Post-Quantum Encryption Standards," news release, August 13, 2024, <<https://www.nist.gov/news-events/news/2024/08/nist-release-s-first-3-finalized-post-quantum-encryption-standards>>, accessed October 22, 2026.

3. Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper, *Transition to Post-Quantum Cryptography Standards*, NIST Internal Report 8547, Initial Public Draft, National Institute of Standards and Technology, Gaithersburg, Md., November 12, 2024, <<https://doi.org/10.6028/NIST.IR.8547.ipd>>, accessed October 22, 2026.

A Closing Thought

When I began my career, identity was largely plumbing: necessary, unglamorous, and rarely discussed in the boardroom. Today it sits at the center of how organizations establish trust, and the arrival of AI is placing it under greater pressure than ever before. That pressure is real, and in places it is uncomfortable. Many of the controls and assumptions we have relied on are being tested, and some will not hold.

I am nonetheless optimistic. The principles that underpin good identity security are sound and well understood. The technology available to implement them, including phishing-resistant authentication, just-in-time access, workload identity federation, continuous evaluation, and AI-assisted analytics, is more capable than at any point in my career. And the community of practitioners working on these problems, across enterprises, vendors, standards bodies, and research, is larger and more engaged than ever.

The question from that conference room years ago will be asked more often, at greater scale, and with higher stakes in the years ahead: who or what is acting, on whose authority, with what permissions, and how would we know if something had gone wrong? My hope is that this book helps you answer it with greater clarity and confidence in your own organization, and that the answers you build will hold up as the enterprise becomes increasingly autonomous.