Managing Identity in Customer Service Operations

By Arynn Crow, Senior Technical Program Manager, AWS Identity
and Jp Rowan, Staff Solutions Architect, Auth0

To comment on this article, please visit our GitHub repository and submit an issue.

Table of Contents

Abstract

This article will establish recommendations for best practices when managing the identities of your end-users in a customer service environment, considering the risks of both external and malicious insider threats. The following recommendations are built from the authors' experiences and observations, and the recommendations included should be considered a starting point to inspire discussion. More rigorous study is necessary to further refine guidelines for this subject.

Introduction

Even in today's highly automated world, there are many jobs that still just need a human. For many organizations, customer service is one of those jobs; when your end users have problems and have exhausted their ability to self-serve, they will turn to your customer service (CS) operations team for support with any number of the services or features you offer. Your CS team is on the frontline and feels your users' pain points more acutely than any other department. Their job, and your users' expectations of them, is to resolve any problem quickly and easily. Therein lies the tension and a core problem for the security-minded identity professional: how do we deliver on our promises of good experience and convenience to our customers while upholding our responsibility to protect their identities?

The cross-section of customer service, IAM, and security is an area that has received comparatively little attention across industry publications and working groups. It is essential to get identity management in CS right due to the consequences for your users and organization for getting it wrong.[i]

Terminology/Glossary

Account Recovery - The process of updating a user's credentials within a scenario where the user cannot validate those credentials
Account Takeover - Account takeover is a form of identity theft and fraud, where a malicious third party successfully gains access to a user's account credentials.[ii]
Agent (also "Customer Service Agent") - The person responsible for communicating with and solving problems on behalf of customers or end-users.
Channel - The communication avenue between you and your end-user, or your agent and their customer. This could be phone, chat, social media, or others.
Credentials - Any attribute or shared secret that can be used to authenticate a user.
Fractured Identity - A case where a single end-user has multiple disparate digital identities.
Impersonation - A scenario where a user is able to perform actions as though they are a known user other than themself.

Knowledge-Based Authentication (KBA) - A method of authentication that uses information known by both the end-user and the authentication service but is not necessarily a secret.
Personal Data - Personal data are any information which are related to an identified or identifiable natural person.[iii]
Social engineering - Social engineering is a method of manipulating people so they give up confidential information, such as passwords or bank information, or grant access to their computer to secretly install malicious software.[iv]
Step-up Authentication - A method to increase the level of assurance (or confidence) the system has regarding a user's authentication by issuing one or more additional authentication challenges, usually using factors different from the one(s) used to establish the initial authenticated session. The need for increasing the level of assurance is typically driven by the risk associated with the sensitive resource the user is attempting to access.[v]
Threat Modeling - Threat modeling is an analysis technique used to help identify threats, attacks, vulnerabilities, and countermeasures that could impact an application or process.[vi]
Username - An identifier unique to the authentication service used in conjunction with a shared secret to authenticate a user.

Why is this different from the rest of my IAM stack?

At first blush, it may be hard to see where customer service – a very operational function of the organization – fits into your otherwise very technical IAM strategy. In fact, CS operations are a critical part of your **IAM strategy,** not only because they represent your organization to customers during important moments ("Why can't I log in to your service my multimillion-dollar business relies on?"), but also because CS operational processes create rich attack vectors for motivated social engineers. We rely on CS agents ("agents" hereafter) to help our customers when they can't help themselves; to ensure their success in this endeavor, we entrust agents with access to private customer data and elevated privileges, from account creation to recovery. Your IAM system could be built with the most secure, sophisticated technology available, but your organization will be perpetually vulnerable unless your CS operational touchpoints are also hardened.

The number of touchpoints between your identity services and your customer service will vary by your type of organization and by the maturity of your organization for automating self-service functions for sensitive account functions. The most common use cases include:

- **General inquiries –** Typically low-risk support requests that do not require modifying account data or divulging personally-identifying information. These requests could include order status updates, troubleshooting, checking balances, etc.

- **Transactional support –** Requests to execute changes on behalf of the account holder, such as making a payment, placing or canceling orders, modifying subscriptions, or adding addresses
- **Account creation and onboarding –** Establishing information about a new administrator or user during account setup, or adding additional delegated users to a "base" account in a nested account schema.
- **Account recovery and state changes** – Highly sensitive requests to restore account access to an end-user, terminate an account, or transfer account ownership to another user
- **Compliance-related requests** - Data Subject Access Requests (GDPR), data deletion requests, Right to Know (CCPA), or similar requests that fall into the scope of a data privacy framework. These operations are sensitive because they deal with potentially large volumes of private customer data, which can result in additional penalties for mishandling.

These use cases likely feel similar to those you must consider elsewhere within your IAM systems, so what makes CS operations different? Your end-users, especially customers, have high expectations about the availability of customer service; the communication channels agents use to interact with customers extend beyond your application stack. Complicating matters further is the reality that the tools you deploy to authenticate and authorize end-users in your web or application environment may be unavailable or impractical to you in a CS environment. Agents often operate across a blend of phone calls, online chats, ticketing, in-person kiosks, social media, and embedded in third-party applications like WeChat. These diverse conditions challenge the application of consistent security rituals like authentication, even if they've been implemented on your online login portal. Organizations looking to preserve both their customer experience and security must weigh the risks of executing functions on the customer's behalf against their relative certainty of a given actor's identity; ideally, this decision-making process should be formalized in an internal framework to ensure decisions are applied consistently and can be inspected.

Establishing Assurance

Key to formalizing a framework for consistency, establishing levels of assurance for the available authentication methods will provide a baseline to determine what types of transactions should be permitted. The concept of assurance levels will likely have already been established as part of the rest of your existing access control policies, but in this case, these levels should be adapted to align with the channels and constraints of your CS interactions. It is likely your customers will have multiple interactions with customer support, and you may track those collective interactions as a "case" or something similar. For the purpose of establishing an assurance level, we will need to look more granularly at the individual interaction, which we will refer to as a "session."

For each session with an agent, the transactions that your agent is allowed to perform should be predicated on the current assurance level. Assurance level will depend on the communication channel or other circumstances but can be increased in a way that your agents are enabled to assist your users without introducing unnecessary risk.

Considering the challenges and constraints that your users will face in a session, it may be necessary to introduce authentication methods that otherwise would not be used for authenticating into your applications or for other self-service workflows.

In comparison to your application stack, it may seem abstract to refer to the process your users are going through in a CS interaction as authentication. In reality, the same primitives can be applied in these scenarios. Designing your authentication methods will help to assess the current assurance level while also reconciling the unique conditions that come into play in a CS session.

Authenticating Through an Application

Your agents and users might communicate through a support portal, contact form, or similar channel directly integrated within your application stack. If so, users will ideally be authenticating through the same service they would for any other application. If that is the case, then authentication and your associated assurance framework should map directly with the actions your agents are allowed to perform.

Authenticating With an Agent

Alternatively, customers may need to rely on an external channel to communicate, and therefore the burden of authentication may fall on your agents. In this case, the goal remains the same, to establish proof the user is who they claim to be.

Notably in the CS experience, some interactions might be very low risk, and it may be acceptable to complete the transaction with an assurance level that would not be acceptable for application access. In contrast, higher-risk operations warrant higher-fidelity authentication methods, or even Step-Up authentication, which requires progressively greater assurance relative to the requested action.[vii] Furthermore, some authentication methods used within a CS interaction may be completely unique to your application stack.

There are many options when it comes to authenticating a user in a customer service interaction. A common theme with all of these authentication methods is the need to create an association with the user’s digital identity ahead of time. Establishing a high assurance level in your customer service sessions requires options tailored to the channels that you communicate through. Those options are only available if you establish the channel or method prior to the session. Creating a secure customer service interaction requires planning and implementation that starts much higher in your IAM stack.

Knowledge-Based Authentication

Knowledge-Based Authentication or (KBA) is possibly the most common, but also the least secure, second-factor mechanism to authenticate your users. KBA involves authenticating a user by asking a set of questions that your user would know the answer to. Common KBA questions include user credentials such as email or username, Social Security Number, date of birth, mother's maiden name, but can be custom or something more specific to the user's interaction with your product. The challenge with knowledge-based questions is that it is particularly difficult to ask a question your user both knows the answer to and that no one else would know the answer to. KBA is hard to store and validate in a secure manner. Unlike a password that can be stored and validated using a one-way hash, KBA answers are typically stored in plain text, which also make them particularly susceptible to being exposed to nefarious actors.

KBA is generally a weak form of authentication, which has been discouraged by the National Institute of Standards and Technology (NIST) in other environments.[viii] It should be understood that having knowledge of a user does not prove they are the account owner or should be entitled to make changes to an account. As an example, children in a household will likely have information about their parents' email addresses, physical addresses, and birth dates, but should not be entitled to access or change information on utility provider accounts their parents own. This information is also sometimes readily accessible online or is a major target for theft (e.g., social security numbers) and is available in criminal databases. Even exposing this information to your agents for the purposes of verification creates a vulnerability. When deciding if KBA is appropriate for some operations in your organization, you should consider the likelihood that the information has already been compromised.

PIN Authentication

PINs and passcodes, like passwords, fall into the category of a memorized secret.[ix]   The intent is to provide similar verification to a password but in a format that can easily conform to a constrained communication channel. PINs can easily be entered into a phone; passcodes can be communicated verbally.

Although PINs can be stored as a one-way hash, due to limited variability in characters they are more easily decrypted. Additionally, if they are provided directly to an agent, PIN authentication suffers similar shortcomings of KBA. As such, whether or not PINs are stronger than methods like KBA is highly dependent on how they are deployed.

Social Authentication

A less obvious option, but valid when leveraging an external application, is to leverage proof of access to the communication channel. In cases where the support channel leverages a social messaging platform (Twitter, WhatsApp, Facebook, Slack), it is possible to access the tool as a form of authentication.

An important step here is that the association of the existing user account and social provider needs to be made ahead of the customer service interaction in order to consider it a valid authentication method.   While a viable option, it is important to consider all the common risks associated with social authentication. Social identity providers do not always verify ownership of email or phone number; they can be created at-will by an end-user and are susceptible to attack outside of the controls of your organization.

Registered Communication Channel (SMS, Phone, and Email) Authentication
Sending a one-time passcode to your user via SMS, email, or phone call is another common method of authentication used to validate a CS session. The code sent to the user is only valid for a single use and should be time-bound; if exposed to your agent or through a man-in-the-middle attack, it does not carry the same risk of being replayed like a memorized secret (KBA, PIN, passwords, etc.).

The use of SMS authentication does suffer some weaknesses. An attacker could gain possession of the user's phone or perform a SIM swap attack.[x] Furthermore, requesting a user to communicate a one-time passcode to an agent normalizes the behavior which could be used as part of a phishing attack. Despite these flaws, SMS continues to be a popular option due to ease of customer use and widespread adoption in application authentication.

Voice Biometrics

Biometrics are increasingly common authenticators across the web, appreciated for their convenience and improved security over methods like password-based authentication. Naturally, organizations have begun deploying these methods in their customer service environments as well, such as by deploying voice biometrics in phone channels to provide low-friction authentication. The appeal of tackling the hard problem of sufficient assurance in customer service with something convenient and secure like biometrics is obvious, but a few challenges you need to consider are:

- There will be different regulatory requirements in each country you operate (or, as with the US, even different regions within the same country may have different requirements). The perception your end users have about biometric ethics may impact the way you collect, store, and apply biometric data. User privacy is paramount.

- If you do not already offer or plan to offer biometric sign-in on your web platform, you're faced with the prospect of building or buying a system only for your customer service channel. In that scenario, you will need to campaign to get your customers to register a biometric specifically for contacting customer service (which they likely hope to never need); alternatively, some organizations "passively" enroll callers into voice authentication.

Biometric implementation in customer service is a complex topic that will require the cooperation of your security, software engineering, and legal teams to ensure you're implementing the correct authenticator for your organization's needs and adhering to all compliance requirements.

Device Authentication

In cases where your users have installed an application on their device, it might be possible to leverage that device as a form of authentication. The most common way is to have the agent trigger a push notification to be sent to the user's device. The service the agent used to trigger the notification then waits for a response back from the device to notify the agent the message has been accepted. This method provides a particularly high level of assurance since it leverages an existing session with your application and proof of possession of the device.

Account Recovery

We are distinguishing account recovery from routine authentication to underscore the increased sensitivity and need for special diligence. Your first goal with account recovery should be for your users to not need it often, as a result of proactive account and security hygiene. Your second goal should be to avoid relying on manual recovery for your customers, such as intervention with customer service, because it is a high-risk operation. Many organizations find that they cannot achieve both objectives and enable their customer service representatives to assist with break-glass account recovery measures when end-users have forgotten or lost access to all their means of authentication, such as by modifying the user's email or password. There are a few common methods of verifying identity when users need assistance recovering their accounts:

- Use of established authenticators previously associated with the account. These methods are strong but may be of limited utility by use case.

- Use of KBA. Even in low-risk use cases, KBA is weak and should be avoided; if this is not possible, bias towards challenge questions that are more extensive than your low-level authentication questions, cannot be obtained online (such as order history

questions known only to you and your customer), and cannot be easily phished from your frontline operations.
- Use of real-world identity documents, such as driver's license and utility bills. If you didn't collect these documents from your user previously for comparison, these should be used in combination with another method to ensure the person who provides you with a document is the correct account owner.

Ultimately, account recovery is a high-risk operation; your users may contact you because they've lost access to any authenticators they could use to self-recover, which means you will be faced with the choice of accepting that your user will be unable to recover their account, or accepting a degradation in your overall security posture. If you maintain a high bar for creating and logging into your accounts, but a weak one for recovering them, this information could proliferate online and be used exploitatively. Always notify your customers about changes to account identifiers and credentials, and give them the option to report, approve, or revert changes initiated with low assurance.

Your organization will need to decide its tolerance for risk in account recovery - or if any risk is acceptable at all - versus its user experience, which may vary depending on what types of accounts you manage. As an example, high-value, high-risk sectors, like large Business to Business accounts, may warrant different processing than retail consumer or public library accounts; there may even be cases where it is appropriate to delegate part of this function to your legal team for more intensive identity verification than your operations will be able to execute.

More on account recovery is available in the IDPro Body of Knowledge article, "Account Recovery."[xi]

Controls

Understanding that your CS operations teams may have access to elevated data and privileges, it is important to have controls in place to prevent misuse (intentional or otherwise) and identify problems quickly. These controls should be considered for all areas within your organization, but there may be additional complexities in organizations with large CS environments.

Permissions controls

In fast-changing environments where seconds matter, operations management will be keen to ensure there is as little downtime for their employees as possible and that the agents have sufficient privileges necessary to perform their jobs. The decision to aim for immediate issue resolution at first contact by assigning extended privileges to the agent is a recipe for overprovisioning. Over time, with insufficient baselining and auditing procedures, this effect can snowball; employees will continue to collect privileges as the demands of their job evolve. Over time (especially in large, complex organizations), the governance conventions of the resources and policies gating those resources shift, leading to role, policy, or attribute explosion, depending on your governance system. This also leads to overprovisioning and, worse, an inability to effectively audit potentially over-provisioned users, as the administrator may not understand what privileges should be removed.

A full analysis of different access control governance models is beyond the scope of this article; other resources, like the IDPro Body of Knowledge "Introduction to Access Control" and "Policy-Based Access Control" offer a more detailed overview of the advantages and disadvantages of Policy-Based Access Control, Role-Based Access Control, and Attribute-Based Access Control.[xii] While the fundamentals of access control do not change for your operations team, depending on the size of your organization, the scale and complexity might; you may find that your operations access needs more drastic and frequent change than sales, engineering, management, et cetera. Finally, it is imperative that your team or IAM resource administrators have mechanisms for auditing privilege use against your organization's policies to ensure your controls are working as intended and preventing misuse.

Risks/Consequences

Administrators of IAM operational functions will, by nature of the job, encounter a number of unique scenarios and edge cases within their organizations beyond what can be fully cataloged in this article. Operations environments can be fast-paced and quick to change, adapting to support the organization as it evolves; nevertheless, it is critical to remain diligent. The channels through which users interact with customer support are desirable attack vectors. Bringing a human into the equation creates the opportunity for exploitation that your application stack would otherwise not be vulnerable to.

The coming paragraphs acknowledge the most common risks, known anti-patterns, and suggested best practices as a reference. This list should not be considered definitive; it is a good starting point to avoid common pitfalls.

Social Engineering

The industry is increasingly acknowledging the significance of the threat posed by social engineers; a 2020 Verizon Data Breach Investigation found phishing and other forms of social engineering were involved in 22% of attacks.[xiii] Customer service agents are especially vulnerable because they are your direct line to the public, they're entrusted with sensitive privileges necessary to resolve tough customer problems, and they likely have a vested, performance-driven interest in making your customer happy. Unmitigated, this can be a severe risk for your organization.

Do:

- **Provide access only to resources that are required to perform the job**. This mitigates damage in case your agent is targeted in a social engineering attack.
- **Routinely educate personnel on the most common types of phishing and engineering attacks.** Ensure they know how to recognize and escalate suspected attacks. Phishing attacks are constantly evolving and becoming more sophisticated; continuous monitoring and updating on current trends is an important part of agent education
- **Establish regular audits of your resources and access rights** to ensure you are continuing to enforce least privilege even as job functions change over time.
- Establish a thorough catalog of the resources your organization maintains and an understanding of their relative sensitivity; require progressively higher-fidelity proofs to gain access to more sensitive resources for both employees and end-users, such as management chain approvals, additional identification, or other checks as appropriate.

Do Not:

- **Use information that is easily accessible to the public** - online or offline - as part of your account authentication or recovery processes

Account Takeover

In a customer service interaction, account takeover is made possible by allowing an attacker to modify a victim's credentials from something the victim knows and has access to, to something the attacker knows and has access to. Credential changes are the catalyst to a chain of events that can result in a valid user losing all access to their account and instead place full control in the attacker's hands. This is the worst case and common result of poor controls within a customer service stack.

It is important to note that credentials can be more than just a password. If a phone number or email address can be used as a channel for account recovery, they too should be considered a credential.

Do:

- Leverage existing authentication methods to establish a secure session with users in customer service interactions. Whenever possible, use existing authentication workflows to establish a legitimate session with your users.
- **Align your session assurance levels with those applied to your applications.** Only when the assurance level matches the requirements for a specific transaction should it also be allowed in a customer service interaction.
- **Leverage existing self-service channels for account recovery when possible.** All self-service account recovery channels should have been threat modeled with the

design of your IAM stack and therefore would not require additional vetting for the purpose of customer service interactions
- **Notify the end-user in the case that a credential has been updated in a customer service interaction.** A message should be sent to all possible (prior) validated communication channels to notify an end-user when a credential has been updated by a Customer Service Agent.
- **Establish controls that allow for changes made by a Customer Service Agent to be reversed by the end-user.** In addition to notification, end-users should have the option to escalate or reverse credential changes enacted against their accounts that they did not authorize.

Do Not:
- **Allow users to update credentials in Customer Service interactions** unless you can satisfy the level of authentication required for these high-risk operations as required by your risk framework

Impersonation

Within an application, impersonation occurs when actions are taken on behalf of a user, without being initiated by that user, are unidentifiable as such. Because customer service agents will often need to perform actions on behalf of other users or possibly replicate another user's experience, it is quite possible that the tools provided to the Customer Service Agents might result in enabling impersonation.

Typically, impersonation occurs because it is simply easier to have a customer service agent login on behalf of the user they are assisting than it is to build out the necessary tooling for them to perform their job securely. Once operationalized, tools and workflows that rely on impersonation create opportunities for users to be harmed without notice and are an enticing target for attackers that wish to wreak havoc without a trace.

Do:
- **Build tools that allow Customer Service Agents to manage end-user data outside of the core application.** Separating the customer service use cases from your core applications makes it easier to audit the actions taken by your agents and helps to avoid scenarios where impersonation might accidentally occur
- **Require end-users confirmation before Customer Support Agents can perform actions on their behalf.** Establishing consent workflows helps build trust with your users and helps to ensure that elevated actions taken by agents are scoped to specific user interactions.

Do not:
- **Allow Customer Service Agents to login as an end-user.** Any scenario where a customer service agent is acting on behalf of an end-user or needs to replicate the

end-user experience must be auditable as such. All actions taken by the agent should be recorded in the system of record as such.

Fractured Identity

Fractured identity occurs when a user is unintentionally associated with multiple accounts. In the case of customer service interactions, this typically occurs when agents establish a new user identity for an existing known user or when a user identity created in a customer service interaction cannot be reconciled with their digital identity. Creating multiple digital identities for an end-user results in a poor end-user experience and can typically result in more overhead expenses wasted to reconcile the fracture.

Do:

- **Create tooling to search for user accounts by fuzzy terms and multiple indexes.** Fractured identities are often introduced when friction is introduced into an agent's workflow and identifying the existing account is more effort than the agent feels is worth the effort. Tooling to find the appropriate user accounts should be implemented with diligence to ensure it aligns with the necessary privacy controls avoiding overexposure of customer data.
- **Create tooling that allows a user to link disparate accounts.** If you have circumstances where fractured account identities might be common, creating self-service tooling to link or merge accounts will save time and minimize frustration for both your agents and customers.

Do Not:

- **Make credentials immutable.** Users will always have justifiable cause to want to update their email, phone number, or username.

Unnecessary Friction

The most secure application is one that doesn't exist. In that vein, it is easy to dismiss the user experience, and therefore any friction incurred by implementing rigorous security controls, as a cost of doing business securely. However, the tradeoff isn't that simple. Bad security experiences have potential risk and financial implications; users who can find workarounds for aggravating security controls will use them. Inefficient processes can also impact your bottom line: every second that your agents spend on the phone or chat attempting to identify a customer is money spent. Review your processes to ensure there are no duplicated steps and verify that there are pathways for customers to authenticate via the same convenient factors they would employ in your web environment (such as hardware authenticators and biometrics).

Do:

- **Match the level of assurance to the risk of the operation.** It may be more appropriate for more onerous authentication processes to start with a basic level of

assurance and use step-up authentication later on if necessary. Deciding which process to use might require you to work closely with your operations teams to categorize different types of actions and assign appropriate authentication methods.
- **Go for stronger proofs instead of layers of weaker proofs.** Delegate as many authentication procedures as possible to something the customer **has** or something the customer **is,** as opposed to knowledge-based authentication, for both security and user experience.

Do Not:
- Pile on authentication layers if they aren't necessary to achieve an appropriate level of assurance for the support your customer needs.

Conclusion
Some concepts from this article may be new to you or instead may offer new ways of looking at and addressing age-old problems for Identity. Because there are likely as many facets to your operations as there are to your business or organization, measures to address their challenges securely won't be one-size-fits-all. It is important to establish a strong partnership between your operations and security teams to solve problems collaboratively. Drawing from the use cases and best practices within this article, as well as other resources within, you will be well-equipped to start these conversations within your organization and begin building or improving a strategy to meet your user needs while protecting their data.

---

[i] As demonstrated by the 2020 Twitter security incident, in which numerous high-profile accounts were compromised, support tooling is a low-complexity vector for high-impact attacks. See: "An Update on Our Security Incident" Twitter, July 2020. https://blog.twitter.com/en_us/topics/company/2020/an-update-on-our-security-incident.html
[ii] "Terminology in the IDPro Body of Knowledge," IDPro Body of Knowledge, accessed 17 April 2021, https://bok.idpro.org/article/id/41/.
[iii] Ibid.
[iv] Ibid.
[v] Ibid.
[vi] Ibid.
[vii] Kaushik, Nishant, "Designing MFA for Humans," IDPro Body of Knowledge, 30 October 2020, https://bok.idpro.org/article/id/49/.
[viii] "NIST 800-63b FAQ". January 2020. https://csrc.nist.gov/publications/detail/sp/800-63b/final
[ix] Paul Grassi (NIST), Elaine Newton (NIST), James Fenton (Altmode Networks), Ray Perlner (NIST), Andrew Regenscheid (NIST), William Burr (Dakota Consulting), Justin Richer (Bespoke Engineering), Naomi Lefkovitz (NIST), Jamie Danker (DHS), Yee-Yin Choong (NIST), Kristen Greene (NIST), Mary Theofanos (NIST). "Digital Identity Guidelines: Authentication and Lifecycle Management," Section

5.1.1.1, National Institute of Standards and Technology Special Publication 800-63B, June 2017. https://csrc.nist.gov/publications/detail/sp/800-63b/final

[x] Barrett, Brian, "How to Protect Yourself Against a SIM Swap Attack," Wired, 19 August 2018, https://www.wired.com/story/sim-swap-attack-defend-phone/.

[xi] Saxe, Dean, "Account Recovery," IDPro Body of Knowledge, 17 April 2021, https://bok.idpro.org/article/id/64/.

[xii] Koot, André, "Introduction to Access Control," IDPro Body of Knowledge, 17 June 2020, https://bok.idpro.org/article/id/42/ and McKee, Mary, "Policy-Based Access Control," IDPro Body of Knowledge, 19 April 2021, https://bok.idpro.org/article/id/61/.

[xiii] "2020 Verizon Data Breach Incident Report" P 7. Gabriel Bassett, C. David Hylender, Philippe Langlois, Alexandre Pinto, and Suzanne Widup. https://enterprise.verizon.com/resources/reports/dbir/2020/introduction/