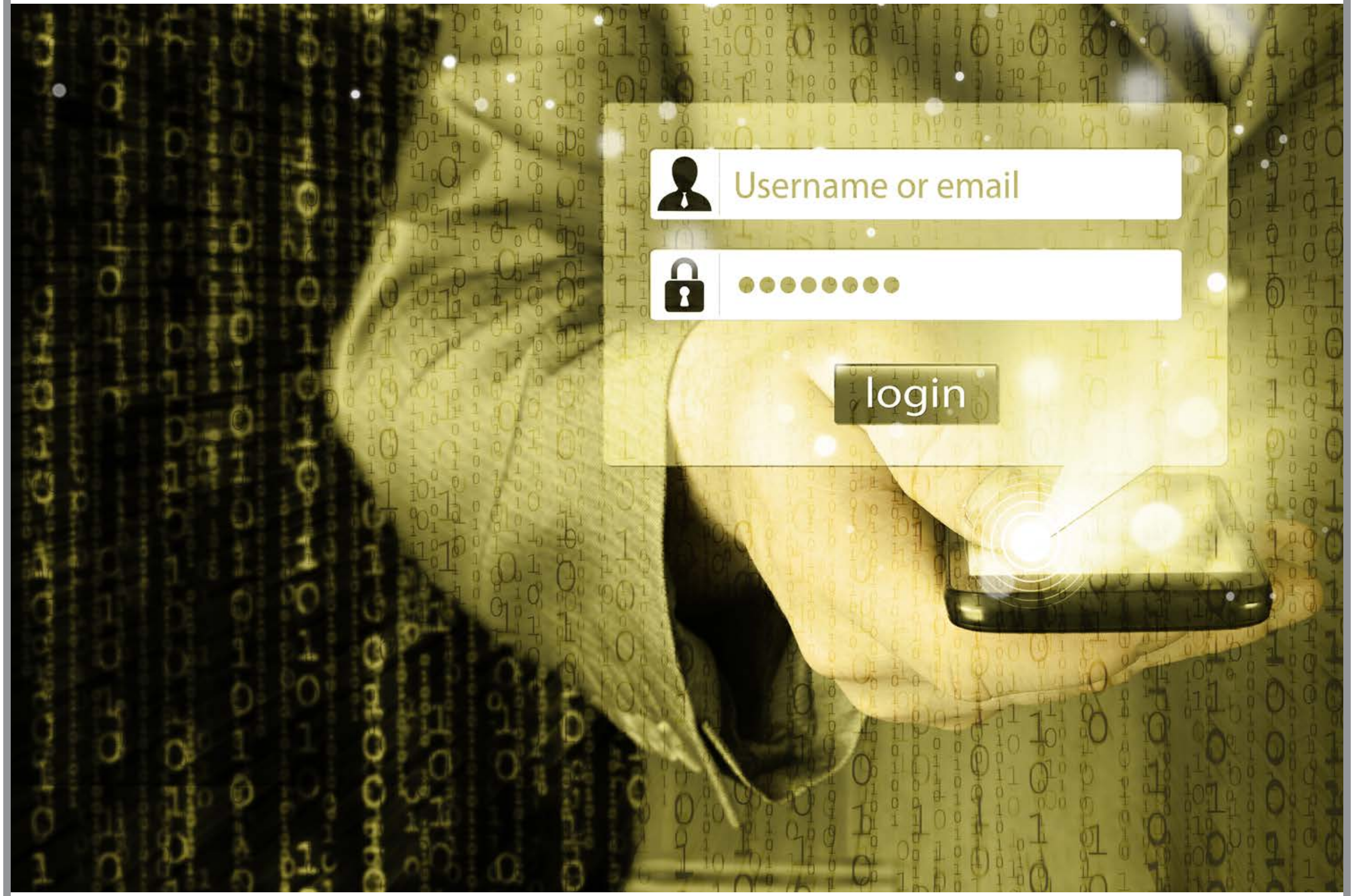
WHITE PAPER

Authentication Choice: The Key to Self-Service Password Reset Adoption



By Nick Cavalancia

SPECOPS

TABLE OF CONTENTS

SPECOPS

I don't think there's one of us who can remember a time in IT when we didn't need to use passwords. Even several decades after the introduction of modern-day computing, with technologies today such as biometrics and hardware tokens, passwords still remain the number one authentication method in 2015—despite their comparative weakness with newer technologies. And although there are these various other techniques and possible replacements for password-based authentication, passwords remain a main-stay for the foreseeable future.

With passwords comes the need for password resets and with that comes a cost.

As long as there are passwords, password resetting is a reality and an IT burden.

As long as there are passwords, password resetting is a reality and an IT burden. Gartner research shows that 40% of support calls are attributed to password resets. This not only takes IT resources away from performing more mission-critical tasks, there's also a considerable cost. When you apply an average cost somewhere in the range of $10 to $30 per support call with an average of 1.5 resets per user per year that's a significant cost for a problem that one might argue is relatively easy to resolve. These calls represent a productivity cost where both the user and the IT professional's time would be better used doing something else. For most organizations, the Helpdesk still remains the primary method for a password reset in 2015.

There's an obvious need here to offload this work. This is where self-service password reset (SSPR) solutions come in. These solutions are intended to allow end-users to reset their own passwords or unlock their own accounts, as such removing this low level task from the IT workload. However it is not uncommon for organizations that have self-service password reset solutions in place to find that their end-users still heavily rely on the helpdesk for password resetting. This should raise some eyebrows, as one has to wonder why users in an organization with a self-service solution in place are bypassing the implemented solution and still using the helpdesk.

As you'll read in this paper, one of the greatest issues revolves around lack of authentication options users have. Traditional methods remain by and large heavily utilized—challenge questions and Mobile Code via SMS—both of which have usability issues.

So, how can you ensure you're truly addressing the password reset problem with an SSPR solution?

Choosing a Self-Service Password Reset Solution

The key to a successful self-service password reset implementation is end user adoption. If users cannot easily enroll and use the solution, it will simply become another piece of shelfware.

How can you know the SSPR solution you choose won't have an adoption problem?

Consider the following criteria when evaluating an SSPR solution to increase the user adoption success rate.

Usability

Like any solution you put in front of users, you need to make sure it's easy to understand, intuitive and simple to use. Otherwise—and you already know this—users are simply not going to use it. An SSPR solution needs to provide a truly easy way for users to access and interact with the system—so much so that it's actually easier than just simply calling the Helpdesk. Some solutions only utilize an internal website for resets. But, if a user cannot log in, how are they supposed to access a website to reset their password?

An SSPR solution needs to provide a truly easy way for users to access and interact with the system.

In the world of today's tech-savvy users, user experience needs to be blended with the security requirements of IT and the organization. Additionally, with so many users working remotely and utilizing cloud-based applications, being able to remotely (and securely) access the SSPR solution is a necessity.

Enrollment

But before you need to worry about usability, you've got to ensure you get your users enrolled in the first place—whether done by the user or by IT on their behalf. This is the step where many implementations fall down. IT organizations get an SSPR solution installed, but either fail to have a communication plan in place which takes engagement across multiple stakeholders (e.g. marketing, senior management) or the solution in place does not have auto enrollment options or only provides limited capability in this area. When the burden lies on the end-user, there is room for disappointing adoption rates. The most successful implementations are those where IT does the enrollment heavy lifting for the user, so that the first time a user needs to use the solution to get their new password they can get access.

Flexibility

Not every organization has the same security requirements around passwords. Think well beyond things like password length and complexity, and focus in on how even within your organization, there are different levels of security. So, when it comes to password reset, for a temp with little access to anything, the simplest of requirements may be necessary for them to reset their password. But for someone with access to financials, intellectual property, personnel, or credit card data, the requirements will likely be vastly different. Having an SSPR solution that can flexibly establish policy based on needs within the organization rather than by providing a single reset methodology is going to be key to maintaining security.

Not every organization has the same security requirements around passwords.

Authentication

This is the most critical of the criteria, as passwords are more than just a field in an Active Directory user account. They represent access to company information, critical applications, and sensitive data. So allowing someone outside of IT to reset a password on their own should be taken a bit more seriously. Stop and think about it for a second—IT is going to put the equivalent of right-clicking a user account in ADUC and selecting Reset Password into the hands of someone other than IT.

But, it's the user doing it themselves, isn't it?

While we think of self-service as always being the user themselves requesting a reset, a proper security stance would dictate that you need to ensure it actually is the user before allowing a password reset. Otherwise, you're implementing a solution that potentially gives inappropriate access to those with malicious intent.

This means the authenticating of the user during the password reset process is probably one of the most attention-worthy timeframes IT needs to be looking at to ensure security. After all, if your user authentication is flimsy, so is the rest of your security.

So, how do you identify whether a person is who they say they are or not?

The Authentication Challenge

When a user calls the helpdesk, it's probably safe to say that there's a simplistic approach to challenging the identity of the person on the other end of the phone. And when you consider the access that password reset will provide, you likely realize challenging a user's claim is probably necessary, especially in larger organizations where the technician has no idea who the user is.

The basic challenge questions that most every solution uses leave much to be desired.

SSPR solutions certainly help, in that they do challenge a user's identity claim, but usually with limited options. The two primary methods of authentication you typically find in SSPR solutions are either a text message or a set of questions provided during enrollment. From a usability standpoint, if SMS latency kicks in, or users simply cannot remember the answers to challenge questions, the user can't reset their password. And, thus, the calls to the helpdesk persist. In general, users need more options that are easy enough for the user, but identifying enough for IT.

From a security standpoint, the basic challenge questions that most every solution uses leave much to be desired. While they have largely progressed well beyond the simple What is your birthdate? or What is your mother's maiden name? (whose answers can usually be found pretty easily on social media), those of you looking for a higher level of security need additional methods to achieve a more certain (and secure) authentication. Just ask Sarah Palin. If you recall, during the 2008 elections, her Yahoo email account was hacked by a simple password reset where all that needed to be provided was her high school and her birthday. Five seconds on Google would provide the answer to those questions. So, challenge questions alone aren't the solution and SSPR solutions need to find a more secure method.

The right answer would be to find a way to balance making it easy for the user with utilizing a method that creates a more secure environment. Many organizations have embraced two-factor authentication, where, in order to authenticate, you need to both know something (usually a password) and have something (such as a token which generates an authentication key). This method certainly helps build up security, but it does little for making it easier for the user (which is a key element in reducing helpdesk calls).

So, is it even possible to implement more secure authentication for password resets that is also more user-friendly?

Making authentication more secure is the easier of the two. But, to meet the need of making it easy for the user, there needs to be a way to log in even if they've forgotten their password. Give them reset options that they already know how to use, and will remember. Like using reusable identities or more simply put identities that they've created or that have been created for them by the organization to log in to other systems.

That way it's not just something they know, but it's also something they already do.

By using additional authenticators, you provide your users with the ability to utilize something they already do to authenticate themselves.

Going Beyond Simple Authentication

The simple answer is to add other authenticators. That is, utilizing other systems or methods that can tell your SSPR "yes that's him/her". Each one can be used to validate the user's identity, and in many cases, more than one is required. By using additional authenticators, you provide your users with the ability to, again, utilize something they already do (such as login to an application, receive a text message, etc.) to authenticate themselves, while IT leverages those authenticators to put layers of security in place, ensuring passwords and, therefore, user accounts remain secure.

The following examples of authenticators are just a few of the possible types available to SSPR solutions today. Some of them fall into the something you know category, others in the something you have, and as you'll see, still another falls into the someone you know category which is probably a new one.

Identity Providers

Identity providers are external applications and systems that participate in a claims-based identity model. The claim (like an email address or user ID) is authenticated by the identity provider who then supplies a digitally signed security token back to the SSPR solution to signify the user has authenticated.

Identity providers can be Enterprise services like Google, Microsoft and SalesForce and even social media accounts like Instagram, Facebook, or Twitter. Now, you might think Social media - really? But for some organizations, this could be a viable option. Take a K-12 school district. They may choose to have students reset their passwords using social media because a) the students have those types of accounts and b) identifying with those providers does establish who the user is. Of course, when we're talking about a traditional organization, its unlikely social media accounts would be viable identity providers.

Even in an organization like yours, services like Google may raise some eyebrows, which initially seems reasonable.

Even in an organization like yours, services like Google may raise some eyebrows, which initially seems reasonable. After all, the concept of taking authentication outside of your organization and placing it in the hands of external applications is somewhat foreign. However, according to analyst firm Gartner, even though most organizations still aren't leveraging social identities due to security concerns a lot more today are and this is primarily due to the fact that many organization are relying on some form of federated identity to enable employees to access web services with a single sign on experience. As the consumption of reusable identities grows and identity providers like Microsoft build market share for their identity services, social identity providers will continue to make security improvements.

However, it's important to keep in mind that the organizations referenced above also utilize adaptive techniques. As such SSPR solutions using identity services need to provide the ability to layer these authenticators on top of each other (by requiring a number of them to authenticate you before your password is reset) so it's not like a user simply logs onto Facebook and they're good. The user has to prove it's themselves by meeting several authentication challenges.

To make this work, Identity Providers and the SSPR solution need to establish what is known as a Federated Trust where one party (in this case, the SSPR) trusts the second party (one of the identity providers) to authenticate on its behalf.

Phone as a Token

Akin to the two-factor authentication tokens, the user's own mobile device either receives or generates a verification code they would provide during reset to further establish the user's identity. In many cases, an SSPR solution would simply send a text to an established mobile device. Note that here is where you can run into latency issues so it's wise to use a service that offers hosted SMS with global coverage. Another phone as a token option that does not rely on SMS is a native application on the device that generates a token.

Peer Authentication

You're trying to reset an Active Directory user account, so why not leverage Active Directory itself even more to authenticate the user. Every user has a Managed By field. By utilizing the account specified in that field and asking that user to assist in identifying that it actually is the user requesting the reset, you speed up the process for the user while eliminating another helpdesk call.

Not every user has the same level of access within the organization.

Putting It All Together

Remember, not every user has the same level of access within the organization. So, it's important to be thinking about the mix of external authenticators in conjunction with the specific security needs of the users within your organization. And add to that the ability to utilize any combination of the authenticators above as needed—think of providing let's say 6 possible authenticators the user can enroll with but only has to authenticate with 3 of them—you've ensured that your users will always be successful in completing the reset process. If one of the authenticators is not available for whatever reason - maybe the user doesn't have their moible phone—the task could still be completed using another one of the five remaining options.

Solving the Authentication Problem

Passwords aren't going away for the foreseeable future. Every single one of them represents a tangible cost within your organization. Helpdesk calls only lower productivity and raise the cost of doing business, all while potentially being addressed in the least secure manner possible—over the phone. And given that use of stolen credentials is the #1 action, according to the 2014 Verizon Data Breach

Investigations Report, it is not only prudent, but necessary, for organizations to take a hard look at what could be the weakest point in their security—resets.

Use of a Self-Service Password Reset solution not only provides a more productive means of resetting passwords, thereby offloading the work from IT, but it also has the potential to do so in a far more secure manner. By utilizing multiple means of authentication—from external accounts, to verification on mobile devices, to even peers within the organizations—as part of the reset process, organizations like yours cannot only make it easier for users to reset their password by taking advantage of accounts and devices they already know, but also make the process more secure, ensuring passwords are only reset when a user is who they say they are.

By utilizing multiple means of authentication as part of the reset process, organizations like yours can make it easier for users to reset their password.

Specops uReset

Specops Software has been addressing password related issues for over 10 years. Based on this this deep rooted foundation the company, which offers both Password Management and Desktop Management solutions based on Group Policy, introduced its next generation self-service password reset solution—Specops uReset.

Specops uReset, builds upon Specops Self-Service Password Reset first generation solution which was developed and honed through years of listening to customers. This insight led to the development of a solution that truly addresses varying security and flexibility needs of organizations through the introduction of flexible multi-factor authentication.

The solution supports over 20 different identity services ranging from social to popular SaaS identities to phone as a token options. The identity services can be layered so IT admins can extend any combination of these to end-users based on role. Admins can also configure different policies, leveraging Group Policy, to dictate how many identity services users need to enroll with versus how many they are required to authenticate with to satisfy the policy. This flexibility means that end-users have options, they are never limited to just one or two forms of authentication which can be limiting and can result in continued calls to the helpdesk. This also translates to increased security by allowing IT admins to either require authentication with more identity services or with higher trust options. To learn more about the solution or to start your free evaluation click here http://www.specopssoft.com/ureset-evaluation/ ■

With nearly 20 years of enterprise IT experience, Nick Cavalancia is an accomplished consultant, speaker, trainer, writer, and columnist and has achieved certifications including MCSE, MCT, MCNE and MCNI. He has authored, co-authored and contributed to over a dozen books on Windows, Active Directory, Exchange and other Microsoft technologies. He has spoken at conferences such as the Microsoft Exchange Conference, TechEd, Exchange Connections, and on countless webinars and at tradeshows around the world.