Conversational
Next-Gen Access: Adaptive SSO and MFA

A ConversationalGeek® Book

Sponsored by Centrify® ZERO TRUST SECURITY

Learn about:

- The problem with passwords and establishing access control
- How SSO and MFA can drive down cost

MINI Edition

By Brien Posey
(Technical Evangelist and Co-Founder of Conversational Geek)

Sponsored by Centrify

Centrify delivers Zero Trust Security through the power of Next-Gen Access. The Centrify Zero Trust Security model assumes that users inside a network are no more trustworthy than those outside the network. Centrify verifies every user, validates their devices, and limits access and privilege. Centrify also utilizes machine learning to discover risky user behavior and apply conditional access — without impacting user experience. Centrify's Next-Gen Access is the only industry-recognized solution that uniquely converges Identity-as-a-Service (IDaaS), Enterprise Mobility Management (EMM) and Privileged Access Management (PAM). Over 5,000 organizations worldwide, including more than half of the Fortune 100, trust Centrify to proactively secure their businesses.

www.centrify.com

Conversational Next Gen Access: Risk Based SSO and MFA (Mini Edition)

by Brien M. Posey

ConversationalGeek

Conversational Next Gen Access: Risk Based SSO and MFAs (Mini Edition)

Published by Conversational Geek Inc.
www.conversationalgeek.com

Trademarks

Warning and Disclaimer

Additional Information

For general information on our other products and services, or how to create a custom Conversational Geek book for your business or organization, please visit our website at ConversationalGeek.com

Publisher Acknowledgments

All of the folks responsible for the creation of this book:

Author:	Brien M. Posey
Project Editor:	Emily Downs
Copy Editor:	Steven Zimmerman
Content Reviewer(s):	J. Peter Bruzzese

The “Conversational” Method

We have two objectives when we create a “Conversational” book: First, to make sure it’s written in a conversational tone so that it’s fun and easy to read. Second, to make sure you, the reader, can immediately take what you read and include it into your own conversations (personal or business-focused) with confidence.

“Geek in the Mirror” Boxes

We infuse humor and insight into our books, both through cartoons and light banter from the author. When you see one of these boxes, it’s the author stepping outside the dialog to speak directly to you. It might be an anecdote; it might be a personal experience.

Within these boxes I can share just about anything on the subject at hand. Read ’em!

SSO Everywhere

Back in the days before PEaaS (Practically Everything as a Service), user authentication was a much simpler matter. Users were given a username and a password and that one account typically gave them the ability to access network resources for which

they had been granted permission. Today, however, a user's domain credentials may gain them access to some applications, but it is just as likely that the user will need a completely separate account to access cloud-based applications.

While the need for a separate account might not initially seem like a big deal, the problem is that, on average, a user may require access to 20 or more applications. That figure is debatable, as some sources report the number of required applications as being much higher or slightly lower, but let's just assume, for a moment, that the figure is correct. If all 20 of those applications were cloud-based, a user could end up having to remember 20 different passwords! That's not even counting the user's main password for logging onto the local network. As if the idea of remembering 20 separate passwords were not bad enough, best practices require long passwords that must be changed frequently.

I think it's probably safe to say that most users are unable and / or unwilling to remember 20 different, frequently changing passwords. Instead, most users end up trying to use the same password for

everything. The problem with this, of course, is that if the password is ever compromised, then whoever knows the password can gain access to all the user's resources.

As bad as that may sound, things are actually just a little bit *worse* than that. That's because users access online resources in their personal life as well. When a user is away from work, they might log on to a social media site, shop at an online retailer, or perhaps do a little bit of online banking. In many cases, users are known to use the exact same password for their personal resources as they use for accessing various cloud applications at work. *This is a huge problem from a security perspective.* If a user were to set up an account on an insecure website, and that website were compromised, the attacker could gain access to a password that could be used to access all the user's personal and work-related resources.

One of the best solutions to the password sprawl problem is to use single sign-on (SSO). Single sign-on takes users back to the days when they only had to remember one password. The user authenticates

into the single sign-on software, and that software takes care of authenticating the users into all their various applications.

As you can imagine, single sign-on capabilities improve the quality of life for the end user. Users no longer have to remember a gazillion different passwords. Furthermore, users may find that they are more productive because they are not wasting time entering passwords every time they switch applications.

Single sign-on software, more importantly, improves the overall security of password-protected resources, because it eliminates the problem of users using the same password to access every one of their resources. Additionally, single sign-on solutions can protect against password theft by disallowing end users from entering passwords into login forms, or prevent the transmission of passwords across the network. While it is true that single sign-on software is based around the use of a single password, the user's password typically cannot be used to gain direct access to the protected resources. The password only works if the

user authenticates through the single sign-on solution. SSO provides users tighter security by granting nuanced control over applications and allowing IT departments greater visibility, enabling monitoring capabilities, and supporting compliance with company policy so users don't have to worry about leaving the company vulnerable to attacks.

The use of SSO also has the potential to reduce *compliance cost*. SSO software makes it easy for IT to determine which employee has access to which cloud applications, and how that access is being used.

SSO software also has the potential to reduce the organization's *operating cost*. According to a 2016 study by Forrester (https://bit.ly/2yak6Pg), password resets cost enterprise organizations an average of about $70 per employee, per year. Additionally, self-service tools have been documented to reduce password reset and account unlock requests by 95%. In some cases, SSO may be able to further reduce costs by allowing application provisioning or deprovisioning through the Active Directory, rather

than requiring each application to be managed individually within the vendor's SaaS clouds.

One more benefit to SSO software, that I never hear anyone talk about, is that it has the potential to increase business agility because new business experiences can be adapted quickly, and with less risk.

Even though running applications in the cloud has only become popular over the last several years, single sign-on software has been in use for quite some time. Out of curiosity, I searched my archives to find out when I first wrote about single sign-on tools. My first single sign-on article was written way back in 2002 – sixteen years ago. The reason why I point this out is because IT has changed a lot in the last sixteen years. Legacy single sign-on software is probably going to be ill-equipped to handle today's access needs.

At the very least, a modern single sign-on solution needs to be able to do two things. *First*, the software needs to support single sign-on regardless of where the application is running. After all, your users will

need to be able to log into applications residing both on premises and in the cloud.

Next, a modern single sign on tool needs to work with any device. You never know if a user is going to be logging in from a PC, a Mac, a smart phone, or some other device. The login process needs to work regardless of the device type that a user may be using.

Adaptive MFA Everywhere

In the last chapter, I talked about some of the ways in which an attacker might steal a user's password. What makes this particular scenario even more troubling is that such an event can go undetected for quite some time.

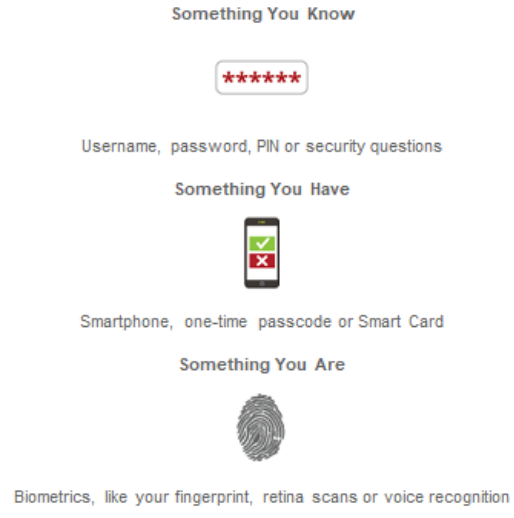
Imagine, for a moment, that a user's password was stolen and that a hacker has remotely logged into your network as that user. Odds are that you would probably never notice the login. Sure, there are some things that could tip you off to the fact that you've got a problem. If, for instance, the user appeared to be logging in from somewhere on the other side of the world when you just saw that user 20 minutes ago, that would be a pretty good indicator that the user's account had been compromised. Likewise, if a user who is barely computer literate were to suddenly launch a sophisticated attack against your domain controller, that's also another clue that you have a problem. But suppose that the hacker just wanted to log in, take a look around, and didn't do anything to send up any red flags. It would probably be impossible to

tell the difference between a hacker logging in using a stolen set of credentials and a legitimate user logging in using those same credentials.

Admittedly, the events that I described in the previous paragraph are a little bit silly and extreme, but they do illustrate an important point. Passwords have a fatal flaw in that *there is no way for you to know, for sure, who is entering the password*. Even if it were impossible to log in to a particular organization's network from outside of the perimeter firewall, password use is still not completely trustworthy. You never really know for sure if the user who just logged on *is* who they claim *to be*, or if it is someone else in the organization who was able to figure out another user's password.

One of the best ways to overcome the weakness of using passwords is to implement a multifactor authentication solution. Multifactor authentication is based on the idea that a user should not necessarily be granted access to the network just because they know a password. Instead, the user will need to jump through at least one more hoop in order to prove their identity. Multi-factor

authentication adds a layer of security as users provide extra information or factors when they access corporate resources. Multi-factor authentication uses a combination of the following factors:



There are lots of different multifactor authentication solutions out there. *We've all used them,* maybe without even realizing that we have used them. For

example, I normally do all my online banking from my desktop computer but, yesterday, I logged on from my laptop. Because the bank did not recognize that particular computer, it sent a text message to my cell phone containing a confirmation code that I had to enter into the online banking website in order to be authenticated.

This is an example of two-factor authentication. The first authentication factor, in this case, was my password. The second authentication factor was my smartphone. The fact that I was able to enter the confirmation code (which had been sent to me by text message) into the bank's website proved that I was in possession of my phone. Since I knew my password and had possession of my phone, the bank assumed that I really was who I claimed to be and, therefore, allowed me to log in.

Multifactor authentication has, occasionally, been described as using a combination of things that you either know, have, or are. You know your password. You have your smartphone, or perhaps a smartcard. The "things that you are" portion refers to biometric authentication such as facial or fingerprint recognition.

Two-factor authentication is not necessarily limited to the use of passwords and smartphones. Smartcards are another popular authentication mechanism. The problem with using smartcards, however, is that they require the use of a smartcard *reader*. That's fine for PCs, but smartphones and other mobile devices are not usually equipped with smartcard readers. There are, however, ways to use derived credentials to achieve smartcard access for mobile devices. In doing so, the mobile device uses a cryptographic credential stored within the mobile device, thereby allowing the device to act like a smartcard.

Risk-Based Access

Tying all a user's accounts to a single identity, and requiring the use of multifactor authentication, can go a long way toward improving the security of your network. Even so, these particular security measures might not go far enough, at least not by themselves. To truly secure your network, you need to base your security model around the concept of zero trust, while also working in risk-based conditional access.

Zero trust is exactly what it sounds like. It's the idea that you should never implicitly trust anyone just because they have access to your network. After all, it's nearly impossible to be 100% certain that the person who is logged in is who they claim to be, and that they are doing what they are supposed to be doing.

You could confirm the user's identity by prompting the user to reauthenticate, but constantly pestering the user to provide their credentials can quickly become cumbersome and counterproductive. It's better to strike a balance between security and productivity. Remember, 99% of the time, the users

are doing exactly what they are supposed to be doing, so you don't want to make them less productive by burdening them with intrusive security.

But what if you could put machine learning and behavior-based analytics to work to improve security, *while also making things easier on the end user*? The idea behind this concept is that a machine learning algorithm watches the user's behavior and, over time, learns what is normal for that user. If the user attempts to perform a privileged operation, and that operation is deemed by the machine learning system to be normal and low-risk, then the security software intervenes on the user's behalf and requests the required privileges so the user doesn't have to. The security is transparent to the end user, and the user does not have to manually request privileges or do anything special.

If, on the other hand, the machine learning algorithm determines that the user is doing something that is a little bit risky, out of the norm, or maybe that the user has just been logged in for a really long time, then the software will step in and

ask the user to reauthenticate into the system. Remember – *zero trust*. We can't automatically assume that the user is who they claim to be, especially if the user tries to do something out of the ordinary. That's why the user receives an authentication prompt. If, however, the user is just going about their business in the normal day-to-day fashion, then the security software keeps the user productive by not nagging them with constant authentication prompts.

The Big Takeaways

Passwords have been the go-to mechanism for authentication for decades. Even so, passwords have outlived their usefulness. Not only are passwords easily compromised, but nearly every cloud resource requires a password. As a result, users are burdened with far more passwords than they can, realistically, remember.

Going forward, authentication will need to be tied to a single identity that represents the user regardless of what resources (cloud or on-premises) they might be accessing. Of course, tying everything to a single identity is not without risk, so some additional safeguards will be essential. Multifactor authentication is a good starting point but, if the security is to truly be effective, the concepts of zero trust and behavior-based analytics will need to be applied.

NOTES

With cyber criminals lurking around every corner it's important to keep your network secure. In this book I will be discussing the different ways to ensure that user truly is who they say they are.

About Brien Posey

Brien Posey is currently in his 4th year of training as a commercial Scientist-Astronaut Candidate, and is preparing for a mission to study polar mesospheric clouds from space. In addition, Posey is a 17 time Microsoft MVP and an internationally published author and conference speaker, with over two decades of information technology experience. You can learn more about Posey's spaceflight training by visiting his Website at www.BrienPosey.com/space.