Conversational

Zero Trust Privileged Access

A ConversationalGeek Book

Sponsored by Centrify ZERO TRUST PRIVILEGE

MINI Edition

Learn about:

- The problems caused by excessive privilege
- How to elevate privileges on an as needed basis

By Brien Posey
(Microsoft MVP, Commercial Scientist-Astronaut Candidate)

Sponsored by Centrify

Conversational Zero Trust Privileged Access (Mini Edition)

by Brien M. Posey

ConversationalGeek®

Conversational Zero Trust Privileged Access (Mini Edition)

Published by Conversational Geek Inc.
www.conversationalgeek.com

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Trademarks

Conversational Geek, the Conversational Geek logo and J. the Geek are trademarks of Conversational Geek®. All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. We cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or programs accompanying it.

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:	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!

The Problem with Privilege

Nearly a decade ago, the so-called Bring Your Own Device revolution began to take hold. Prior to that, access to corporate networks was almost exclusively confined to domain-joined, tightly controlled PCs. However, the proliferation of consumer electronic devices such as tablets (most notably the iPad) and smartphones made it nearly impossible for IT pros to ignore the demands, made by users and executives alike, to use personal devices for work.

Over the last several years, we have seen countless Fortune 500 companies, government agencies, and non-profit organizations get hit with security breaches. Naturally, security spending has skyrocketed, seeing as nobody wants to be mentioned as the latest casualty in tomorrow's headlines. Despite all the extra attention that is being given to IT security, however, there is one aspect which is often overlooked – *privileged accounts.*

Privileged accounts are the accounts that are used to perform various administrative tasks, such as setting up new user accounts, resetting a user's password, or granting a user access to an application. As essential as it is for the IT department to be able to perform these and other administrative tasks, privileged accounts pose serious risks to an organization's security.

One of the problems stemming from the use of privileged accounts is that of anonymity. I'll explain. Operating systems, such as Windows and Linux, come with a built-in Administrator or Root account that is used to perform administrative tasks. If the IT staff were to share this account, then it becomes impossible to trace an administrative action back to the person who performed the action. The administrative action could have been performed by anyone who had access to the account.

I have actually run into this problem in real life. I once worked for an organization in which an administrator had gone rogue. Because the entire IT staff shared the built-in admin account, it was impossible to know for sure who was sabotaging the organization (although we all had our suspicions). The situation was only resolved when the IT staff was locked out of the built-in account and forced to use individual accounts.

The problem of anonymous administrative activity is relatively easy to resolve by creating separate accounts for each IT staff member and taking steps to prevent the IT staff from using the built-in account. Even so, this approach solves one problem, but creates another.

Imagine, for a moment, that a particular organization has five administrators and that, rather than sharing an admin account, each of the five administrators has been given their own unique admin account. In addressing the problem of administrative anonymity, the organization has increased the size of its attack surface. Now, instead of having one privileged account that could potentially be compromised, there are six; the five individual accounts, *plus* the built-in account.

The point is that while administrative anonymity certainly can be a problem, it is only a symptom of a much larger problem. Simply put, the accounts that IT staff members use each and every day have typically accumulated a very broad set of privileges.

So why is this a problem? Imagine what would happen if an IT staff member's account were to be compromised. The hacker would gain the same level of privileges as the person whose account they were using. If, for example, the IT staff member had the ability to create user accounts, then the hacker might set up their own "back door" user accounts that could be used to log into the network in a less-suspicious way. If the IT staff member has access to the network's security logs, then the hacker would also have access to the security logs and might use that access to cover their tracks.

A privileged account does not necessarily have to be hacked to become a problem. If a user contracts a malware infection, the malware (including ransomware) operates under the same set of privileges as the user who is logged in. In other words, *the more privileges that the user has, the more damage the malware could potentially do.*

There are other reasons an accumulation of administrative privileges is bad. I have heard stories of IT pros amusing themselves on the weekends by snooping around in the Payroll database to see how much money their coworkers make. I have also heard a couple of stories of IT pros using information from the HR system to cyberstalk someone in the organization.

Privilege Elevation on an As-Needed Basis

Administrative privileges present something of a paradox. On one hand, administrative privileges are absolutely required to perform a wide variety of administrative tasks. On the other hand, these same privileges pose a significant risk to the organization's security and general wellbeing. Thankfully, there is finally a realistic solution to this age-old problem thanks to the concept of elevation on an as-needed basis.

The idea of elevation on an as-needed basis can exist in many different forms. If you want a really simple example, look no further than PowerShell. Microsoft provides a basic PowerShell interface with limited capabilities, but also gives you the option of running PowerShell as an administrator.

Elevation on an as-needed basis is something of a generic form. There are many different examples of impromptu privilege elevation, but the two that seem to really be gaining traction lately are called *Just Enough Privilege* and *Just in Time Privilege*.

Just Enough Privilege

The concept of Just Enough Privilege is based on the idea that, in the world of IT, administrative responsibilities are becoming increasingly specialized, making general purpose privileges really undesirable.

Microsoft does a good job of explaining the concept of Just Enough Privilege in a PowerShell blog post that you can find at: https://docs.microsoft.com/en-us/powershell/jea/overview . Borrowing an example from that article, imagine that an organization has grown to be large enough that generalized administration has become impossible. Instead, the organization has hired IT staff members to handle specialized areas of administration. The organization might, for instance, have an Active Directory administrator, an Exchange Server administrator, a Web services administrator, and the list goes on.

So, with that in mind, also imagine that the organization has hired an administrator who is dedicated to maintaining the company’s DNS servers. In a Windows Server environment, DNS servers are not used solely for resolving Internet domain names; the Active Directory is also completely dependent upon DNS. As such, it is relatively common for the DNS services to be installed on Active Directory domain controllers.

Now, imagine that the DNS administrator needs to log on to the DNS server (in this case, a domain controller) to fix a problem that cannot be corrected through the DNS Management Console. *Do you see the problem?* An administrator who is tasked only with managing the organization’s DNS infrastructure has to have the ability to log on to a domain controller.

The concept of Just Enough Privilege directly addresses this problem by dividing management responsibilities into a series of roles. Each role is assigned the permissions necessary for completing the tasks associated with that role and nothing more. This concept is sometimes referred to as *role-based access control*. In the case of the DNS Server administrator, the principles of Just Enough Privilege would allow the admin to log on to the DNS server to perform DNS-related configuration tasks, but would prevent the administrator from tampering with the Active Directory.

Just in Time Privilege

The concept of Just in Time Privilege is relatively new to the world of IT. It is based on the idea that, even if a user requires administrative privileges, they might not need those privileges at all times. Hence, Just in Time Privilege allows you to set an expiration date for administrative privileges.

Suppose that a network administrator needs to create new user accounts for some recently hired employees. The old way of doing things would have been simply to grant that administrator the necessary permissions to be able to create new accounts on an as-needed basis. With Just in Time Privilege, however, that privilege only exists when the administrator explicitly asks for it.

There are a few different implementations of Just in Time Privilege, and they all work a little bit differently. Typically, however, there is some sort of approval workflow associated with the request for privilege. The administrator might, for example, log into an automated system and indicate that they need to set up twenty new user accounts. The automated system might then respond by saying, *OK, you have been granted permission, but only for the next two hours*.

The nice thing about Just in Time Privilege is that it limits the privileges associated with otherwise privileged accounts. In other words, the accounts used by members of the IT staff would be treated as standard user accounts, except in very specific circumstances in which elevated privileges are required. The benefit to this approach is that, if a privileged account were to be compromised, the account would be practically useless to the hacker because it does not have any permissions associated with it.

Zero Trust Privilege

In the previous section, I introduced the concept of Just in Time Privilege. On the surface, it may seem as though this concept would work really well for some organizations, and yet be completely impractical for others. *I can see why you'd think that*.

When I introduced the topic of Just in Time Privilege, I borrowed an example for Microsoft in which an organization had hired an administrator to do nothing but manage DNS records. Because this administrator works in the DNS Management Console (or uses DNS-related PowerShell cmdlets) all day long, it probably seems completely counterproductive to apply Just in Time Privilege to that user.

Given the way that Just in Time Privilege usually works, I would completely agree with you if you said that the use of Just in Time Privilege might not be the best fit for a dedicated DNS administrator. However, there are next-generation Just in Time Privilege solutions that do not force an administrator to say "mother may I" every time they need to perform an administrative task.

Before I get into a discussion of how such a solution might work, there is one more concept that I need to introduce. This is the concept of Zero Trust. At its simplest, Zero Trust means that you don’t automatically trust someone just because they are on your network.

Imagine, for a moment, that a user logged in remotely and then tried to perform some administrative activity. Even if that activity was within the scope of what the user is normally allowed to do, it may not be in your best interest to trust that activity. After all, the user has been logged in remotely, although perhaps using a password which can easily be stolen. They may be who they claim to be, but only through closer inspection of their normal behavior, the risk level of the request, and whether they have been authenticated using MFA should we trust their request.

If an organization hopes to truly keep its network resources secure, it needs to find a way to incorporate principles such as Just Enough Privilege, Just in Time Privilege, and Zero Trust. The trick is to do it in a way that does not become intrusive or make the end user counterproductive. The key to achieving this is to use machine learning to learn what is normal behavior for each user.

Here are some more hypotheticals... a particular user comes into the office every morning, logs onto the same desktop computer, and gets to work reviewing the security logs from the night before. If a machine learning algorithm determines that this particular user logs onto a machine with this specific MAC address and IP address at 8 o’clock in the morning on weekdays, and begins browsing the security event logs, then it could easily establish that this is normal behavior for this particular user. If, on the other hand, an administrator who normally only works with the organization’s DNS servers logs in at 3 o’clock in the morning from a computer located in a rogue nation, halfway around the world, and begins poking around in your security logs, it’s probably malicious activity. The machine learning algorithms should be able to pick up on that and request additional factors of authentication or shut down the activity, and alert you to the problem.

Of course, these are extreme examples. Zero Trust comes into play when an administrator attempts to do something that they are allowed to do, but something about the action is a little bit outside of the norm. Maybe the user had been working on something else for a while, or perhaps they log on from a different PC than normal, or maybe this particular action isn’t really tied to their normal administrative functions.

If a system were designed around the privileges, as they had been defined in the past, then such actions would most likely go completely unnoticed. Yet with a machine learning algorithm, which can determine that the privilege is being leveraged, the administrative action may receive some extra scrutiny.

A machine learning algorithm might, for instance, determine the level of risk based on the type of action that is being attempted, and how unusual the action seems to be. If the risk is deemed to be excessively high, then the system can prompt the user to complete some sort of challenge. Typically, this would probably be a prompt for multifactor authentication; after all, you would not want the system to simply ask for the user’s password. If the password had been compromised, the hacker performing the action would presumably know the password.

The Big Takeaways

IT has seen massive changes in recent years. These changes have rendered tried and true security measures all but obsolete. Fortunately, new technologies exist that can help facilitate a least privilege model. This approach allows administrative users to use a single account (as opposed to using one account for day-to-day tasks and another for administrative tasks), while also eliminating most of the risk associated with that account.

NOTES

Users need to have the correct amount of access to properly perform their job, but what happens when a user has too much access. Join the conversation as I discuss tips for ensuring users have just the right amount of access leaving your network secure.

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.

ConversationalGeek®