Deep-Dive Analysis of Multifactor Authentication Fatigue Attacks

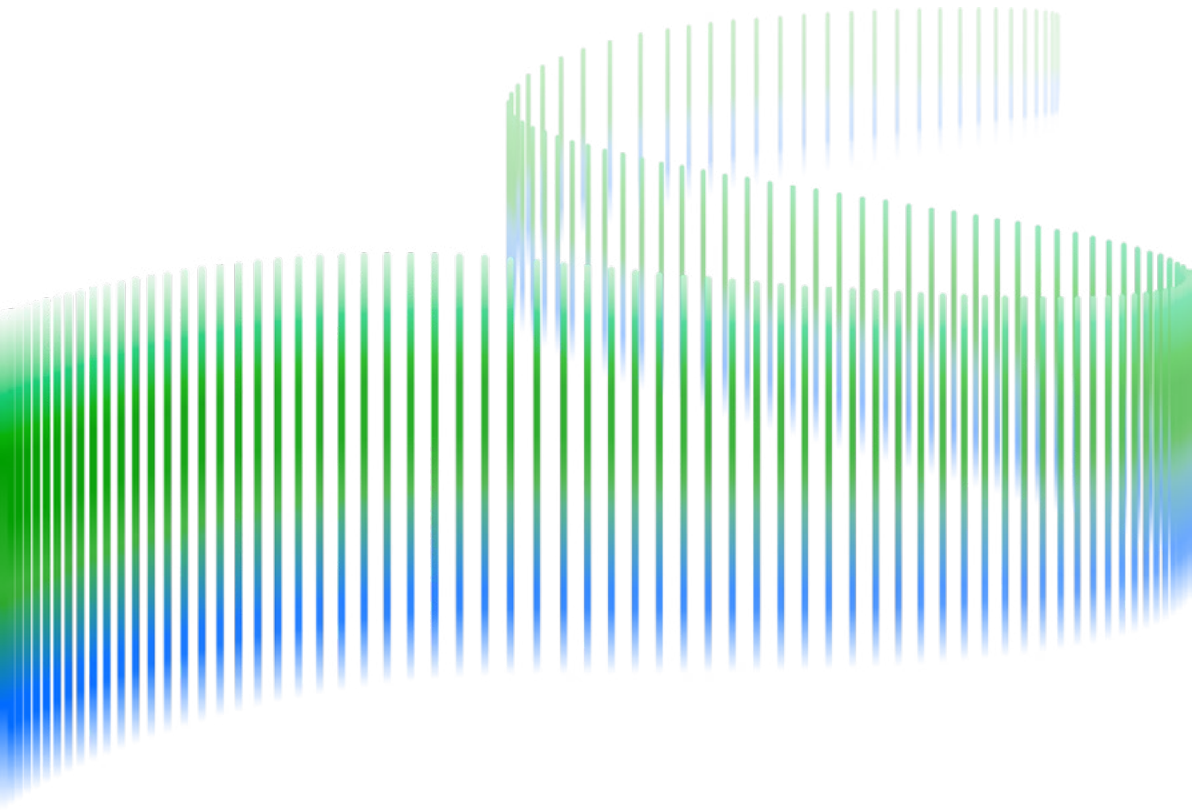


Table of Contents

Introduction

Nearly every cyberattack involves stealing credentials that are then used to gain initial access, maintain stealth and persistence, or conduct lateral movement to further access data, applications, or systems across an organization. Simple username and password combinations are easily transportable and reusable, making them a consistent target for cybercriminals.

To reduce the risk of compromised user accounts, multi-factor authentication (MFA) tools require users to pass two or more authentication tests when logging into applications. This basic security measure adds complexity for cybercriminals, making it far more difficult to misuse stolen credentials. Cybercriminals use many tactics to defeat MFA security measures, such as SIM swapping, transparent proxies, and brute force attacks. One frequently successful method in real-world attacks is a social engineering tactic known as MFA fatigue.

This white paper will cover how cyberattacks have evolved from simple credential theft into MFA bypass attempts, explain MFA fatigue and how it functions, share examples of high-profile attacks, and provide guidance for detection and mitigation using a **security information and event management (SIEM) platform**.

Reduce Risk of MFA Fatigue Attacks

As with any unusual or abnormal behavior on a network, if you know what to look for and which log data to collect, you can spot attacks while they are occurring and stop them.

Exabeam Fusion is a cloud-native SIEM that provides visibility into user and entity behavior involving critical data and infrastructure . Self-hosted LogRhythm SIEM, in conjunction with LogRhythm Intelligence™ is the hybrid solution. Both solutions help security teams detect, investigate, and respond to cyberthreats. Look for Exabeam insights throughout this paper.

History of Credential Harvesting

Attacks on MFA Started with Simple Credential Harvesting

Cyberattacks generally begin with active reconnaissance techniques, where threat actors research potential targets to identify weaknesses or vulnerabilities that might be exploited. For instance, a threat actor may conduct network and port scanning to identify an organization's IP addresses, active hosts, running services, and more.

Furthermore, to gain initial access to resources, a threat actor might initiate social engineering tactics to harvest user credentials. For example, threat actors often target employees via email by asking them to reset their passwords. These fake emails can look legitimate, but the link included redirects to a fraudulent website designed to collect user credentials. This type of attack technique is not uncommon, but it requires resources and skills that some unsophisticated cybercriminals may not possess. This is where the tactics of initial access brokers (IABs) have become a component of the cyberthreat landscape.

IABs are threat actors who specialize in infiltrating computers and networks to obtain valid credentials and then sell unauthorized access to other malicious actors. For example, IABs help threat actors using Ransomware as a Service to speed up their attacks by providing initial access to victim networks via a purchased set of compromised credentials. IAB services assist cybercriminals of all skill sets with the initial access required to launch attacks.

Approximately 380 access brokers sell credentials on the dark web, with prices ranging from **five dollars to hundreds of thousands of dollars**. Since this is such a lucrative business, IABs sell harvested credentials on the dark web at a rate of **10,000 a month**. The impact of this growing market is felt worldwide:

- **59% of organizations** have experienced phishing-based campaigns focused on stealing credentials.
- **23% of organizations** have experienced brute-force password attacks.
- **83% of organizations** experienced an identity-related breach involving compromised credentials.

Organizations have countered this challenge by implementing stronger identity and access management (IAM) processes, such as MFA technology, to address this challenge.

Organizations Respond by Shifting to MFA

Nearly 99 percent of organizations use MFA to some degree. Two-thirds only have MFA in use by specific users with access to sensitive data, while **one-third has almost everyone in the organization using MFA**. Even Microsoft defaults to MFA within Microsoft 365; according to the organization, **MFA reduces the risk of compromise by 99.2%**.

Although MFA tools significantly reduce risk, cybercriminals continually find creative ways to work around this extra layer of defense. It is essential to be aware of these techniques and understand how to detect potential threats before they become damaging breaches.

MFA Fatigue: What It Is and How It Works

Threat actors may attempt to bypass MFA mechanisms and gain access to accounts using the MFA fatigue tactic. This method of authentication attack is also known as Prompt Bombing. It occurs when an attacker obtains a valid username/password combination for a network that has push notifications as part of its MFA solution. Often via automation, an attacker sends large volumes of push notifications to a victim's MFA device. Eventually, the victim approves one of the push notifications, which allows the attacker to log in.

MFA fatigue is a technique both script kiddies and Advanced Persistent Threat (APT) actors use. It is effective because threat actors do not need exceptional technical skills to spam a user with MFA requests once valid credentials are acquired.

MITRE ATT&CK® Definition

The MFA fatigue technique is mapped to the ATT&CK framework and labeled as, ***Multi-Factor Authentication Request Generation (T1621)***. ATT&CK defines the technique as "Adversaries may attempt to bypass multi-factor authentication (MFA) mechanisms and gain access to accounts by generating MFA requests sent to users."

The next section will dive into real-world examples of this attack.

MFA Fatigue Attack Examples

MFA Fatigue in Action: Uber

In 2022, a threat actor associated with the Lapsus$ hacking group allegedly **infiltrated Uber's virtual private network (VPN)**. They used stolen credentials to repeatedly push authentication notifications to the credential owner, who eventually accepted the approval request. This granted the threat actor access to internal Uber systems like Slack. They then accessed a shared drive that contained PowerShell scripts, from which they pilfered administrative credentials to Uber's privileged access management (PAM) service.

MFA Fatigue in Action: Cisco

In 2022, a threat actor **infiltrated Cisco via a Google account** using an MFA fatigue attack and adding a new MFA device to the account, giving them persistent access. They moved laterally to gain administrative access and installed tools like TeamViewer, Mimikatz, and Cobalt Strike. These actions can be precursors to a ransomware attack. Luckily, Cisco's security team identified and mitigated the access before serious damage occurred.

What Does MFA Fatigue Look Like?

The next section will show examples using log data from Okta Identity Engine to better prepare an organization to identify MFA fatigue. The use of this platform is in no way meant to imply that it is less secure or more prone to MFA fatigue attacks; in fact, because the weak link in MFA fatigue is the user, any authentication platform or service is equally prone to these types of attacks.

Below, explore four specific logged actions using the Okta log data. Remember that the particular log data will change based on the identity service providing the logs.

Successful Active Directory (AD) Authentication and Push Verify

This logged action typically represents a successful login leveraging MFA. As shown below, using the ***Action Type*** column as our guide, the user authenticates against Active Directory (**user.authentication. auth_via_ AD_agent** with a ***Result Message*** column value of ***Success***) and verification of the authentication (meaning a successful submission of username and password) is confirmed (**user. authentication.auth_via_mfa** with a Result Message column value of ***Success***).

Next is the successful MFA push request (**system.push.send_factor_verify_push**) and then a successful push verification (***Action Session Type*** of **OKTA_VERIFY_PUSH** with an ***Action State*** of **OV_RESPONSE_ APPROVE**).

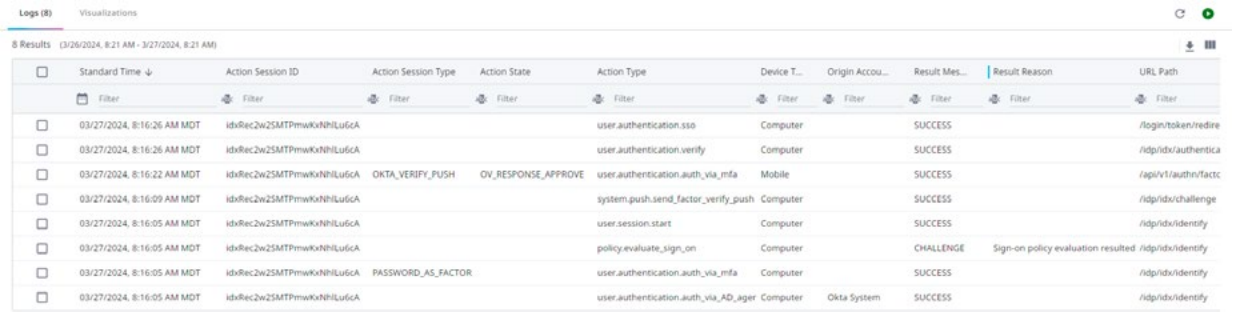


Figure 1. Successful AD Authentication and Push Allow

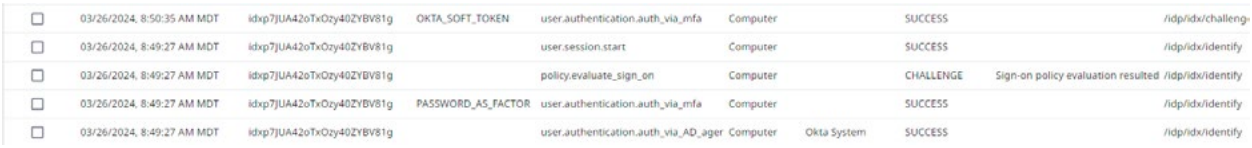
The successful **user.session.start** and **user.authentication.verify** logs indicate the user has been authenticated and a session is being established.

Figure 2. Successful AD Authentication and Push Deny

Successful AD Authentication and Push Deny

The image above has a log entry with an ***Action State*** of **OV_RESPONSE_DENY**. This entry represents when the credential owner has denied an MFA request. If repeated push denies for a given logon session are observed, this means the "fatigue" portion of the attack, where a logon attempt is repetitively made, resulting in an equal number of MFA push requests that the credential owner denies.

When multiple authentications with failed deny push log entries are followed by a successful push verification, it represents a successful wearing down of the credential owner and a push verification of their logon request, granting the threat actor access.



☐	03/26/2024, 8:50:35 AM MDT	idxp7JUA42oTxOzy40ZYBV81g	OKTA_SOFT_TOKEN	user.authentication.auth_via_mfa	Computer		SUCCESS		/idp/idx/challeng
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g		user.session.start	Computer		SUCCESS		/idp/idx/identify
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g		policy.evaluate_sign_on	Computer		CHALLENGE	Sign-on policy evaluation resulted	/idp/idx/identify
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g	PASSWORD_AS_FACTOR	user.authentication.auth_via_mfa	Computer		SUCCESS		/idp/idx/identify
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g		user.authentication.auth_via_AD_ager	Computer	Okta System	SUCCESS		/idp/idx/identify

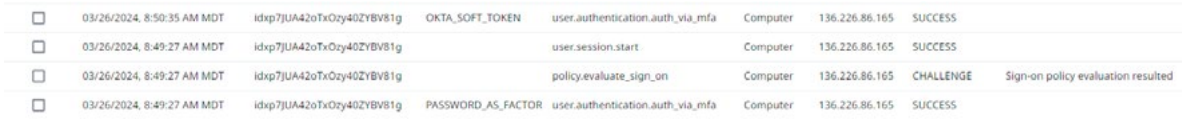
Figure 3. Successful AD Authentication and OTP Code Entry

Successful AD Authentication and Successful One-Time Password (OTP) Code Entry

This logged action represents when a user decides to type in a code rather than responding affirmatively to a push verification. In this case, there will be no log entries related to push verifications and (if applicable) push denies; rather, in the case of the Okta logs, two **user.authentication.auth_via_mfa** logs will be observed. One with an ***Action Session Type*** of **PASSWORD_AS_FACTOR**, indicating the user entered their password correctly, followed by an ***Action Session Type*** of **OKTA_SOFT_TOKEN** with a ***Result Message*** of **SUCCESS**.

Exabeam Insights: Machines in MFA Are Meaningful

When trying to determine whether an authentication is malicious or not, the machines used provide specific insight. First, is the obvious determination whether the machine being used to authenticate a specific credential is one that has been used previously. But second, as shown below, seeing that an OTP was provided on the same computer (based on IP address) indicates that the user of the credential is obtaining the OTP likely on a mobile device or hardware key and typing it into the same system where they provided their credentials.

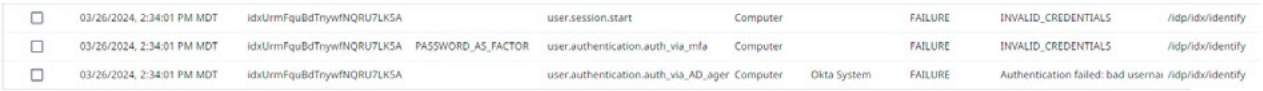


☐	03/26/2024, 8:50:35 AM MDT	idxp7JUA42oTxOzy40ZYBV81g	OKTA_SOFT_TOKEN	user.authentication.auth_via_mfa	Computer	136.226.86.165	SUCCESS	
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g		user.session.start	Computer	136.226.86.165	SUCCESS	
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g		policy.evaluate_sign_on	Computer	136.226.86.165	CHALLENGE	Sign-on policy evaluation resulted
☐	03/26/2024, 8:49:27 AM MDT	idxp7JUA42oTxOzy40ZYBV81g	PASSWORD_AS_FACTOR	user.authentication.auth_via_mfa	Computer	136.226.86.165	SUCCESS	

This means the device providing the OTP is in the possession of the credential owner, indicating that this is a valid logon.

Unsuccessful AD Authentication

To be thorough, it is essential to show the logging activity in cases where the threat actor does not have credentials and is attempting to brute force an account. As shown below, a failed logon attempt is observed with an ***Action Type*** of user.**authentication.auth_via_AD_agent**, a ***Result Message*** of **Failure**, and a ***Result Reason*** of **Authentication failed: bad username** or **password**. Observing this repeatedly with the same Target Account value would indicate a brute force attack.



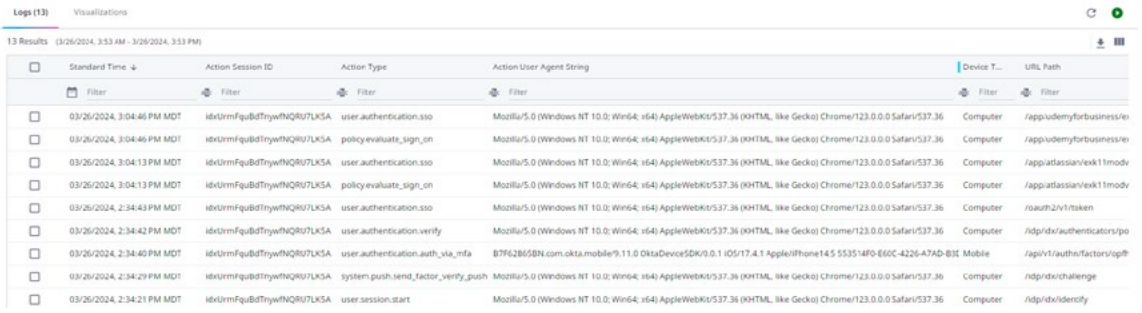
☐	03/26/2024, 2:34:01 PM MDT	idxUrmFquBdTnywfNQRU7LK5A		user.session.start	Computer		FAILURE	INVALID_CREDENTIALS	/idp/idx/identify
☐	03/26/2024, 2:34:01 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	PASSWORD_AS_FACTOR	user.authentication.auth_via_mfa	Computer		FAILURE	INVALID_CREDENTIALS	/idp/idx/identify
☐	03/26/2024, 2:34:01 PM MDT	idxUrmFquBdTnywfNQRU7LK5A		user.authentication.auth_via_AD_ager	Computer	Okta System	FAILURE	Authentication failed: bad usernai	/idp/idx/identify

Figure 4. Failed AD Authentication

Exabeam Insights: Focus on the Data That's Important to You

Network, system, device, and other similar logs contain many details. To properly focus on what is essential to cybersecurity analysts, threat hunters, or other investigators, it is necessary to identify specifics critical to the analysis.

This presentation of log data includes the ability to select specific fields for better visibility. For instance, in the analysis of the logon behavior below, the Session and User Agent fields were added to not only highlight the multiple actions within the same session but also identify the devices involved.



Logs (13) Visualizations

13 Results (3/26/2024, 3:53 AM - 3/26/2024, 3:53 PM)

	Standard Time	Action Session ID	Action Type	Action User Agent String	Device T...	URL Path
	Filter	Filter	Filter	Filter	Filter	Filter
☐	03/26/2024, 3:04:46 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.authentication.sso	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/app/udemyforbusiness/e
☐	03/26/2024, 3:04:46 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	policy.evaluate_sign_on	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/app/udemyforbusiness/e
☐	03/26/2024, 3:04:13 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.authentication.sso	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/app/atlassian/exk11modv
☐	03/26/2024, 3:04:13 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	policy.evaluate_sign_on	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/app/atlassian/exk11modv
☐	03/26/2024, 2:34:43 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.authentication.sso	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/oauth2/v1/token
☐	03/26/2024, 2:34:42 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.authentication.verify	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/idp/idx/authenticators/po
☐	03/26/2024, 2:34:40 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.authentication.auth_via_mfa	B7F62B65BN.com.okta.mobile/9.11.0 OktaDeviceSDK/0.0.1 iOS/17.4.1 Apple/iPhone14,5 553514F0-E60C-4226-A7AD-B3E	Mobile	/api/v1/authn/factors/opft
☐	03/26/2024, 2:34:29 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	system.push.send_factor_verify_push	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/idp/idx/challenge
☐	03/26/2024, 2:34:21 PM MDT	idxUrmFquBdTnywfNQRU7LK5A	user.session.start	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	Computer	/idp/idx/identify

Detecting MFA Fatigue

A few ways to identify MFA fatigue are outlined in the ATT&CK framework's coverage of this attack technique.

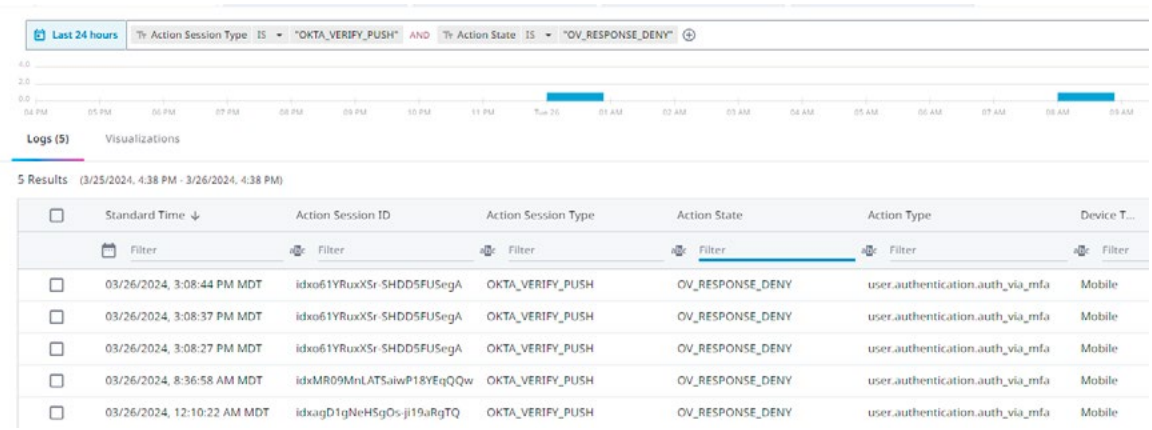
Repeated Push Denies

One of ATT&CK's recommended detections looks for an unusually large number of push denies that could indicate a user's credentials have been stolen. Usually, an analyst would not see more than one or two, if any, push denies from users legitimately authenticating. However, threat actors often run scripts that generate many more pushes than a person could accomplish manually, with each one either timing out or being denied.

Exabeam Insights: How Many Push Denies is Too Many?

While one to two push denials may seem acceptable, security teams should identify a value that users in the organization would consider unusual or abnormal. To determine this value, it is necessary to identify how frequently push denies occur within the organization and what would be regarded as "normal" versus "abnormal" activity.

New-Scale SIEM easily enables the visualization of specific events over a particular period. As shown below, the ***Action Session Type*** (**OKTA_VERIFY_PUSH**) and ***Action State*** (**OV_ RESPONSE_DENY**), along with the desired timeframe of 24 hours are used as criteria to search for and observe related activity.



In general, there are two specific scenarios to consider. The first is simply repeated push denies for a single user, representing a failed MFA fatigue attempt that does not result in the threat actor gaining access. The other scenario is repeated push denies followed by a push allow for the same user, representing a potentially successful MFA fatigue attack that results in a threat actor gaining credentialed access.

Rapid Logon Attempts

This detection ignores the number of login attempts and instead focuses on the frequency of logins within a specific period of time. Again, considering that malicious logon attempts will be scripted, they will likely occur in rapid succession. As with the previous example, an organization would need to determine what might be considered "normal" in its environment. A threshold count of five or ten logons is modest, given that a script-based attack may generate hundreds of attempts per minute.

Location-Based Mismatches

This type of detection is a bit trickier to implement, as it is based on observing a mismatch in the logged location during the logon attempt and the associated push response. This method of detection has some challenges.

The first challenge is the public IP address determines that location. Depending on the network scenario, public IPs can change quickly and dramatically. For example, an employee works for a company based out of London, but the employee lives in Denver, Colorado. Some of the log activities generated by the employee might result in a Denver public IP. However, should the employee connect to the corporate VPN terminating in London, the logging activity changes to a London-based IP.

Another challenge is a typical example where an employee logs onto the company network via their laptop and then uses their mobile phone to perform MFA push notifications. If their mobile phone is not connected to the WiFi, but the laptop is connected (as may be the case if they are working from a coffee shop and only connected their laptop to the free WiFi, leaving the mobile to connect to the Internet via their mobile provider), their authentication will show two different devices and, potentially, different locations

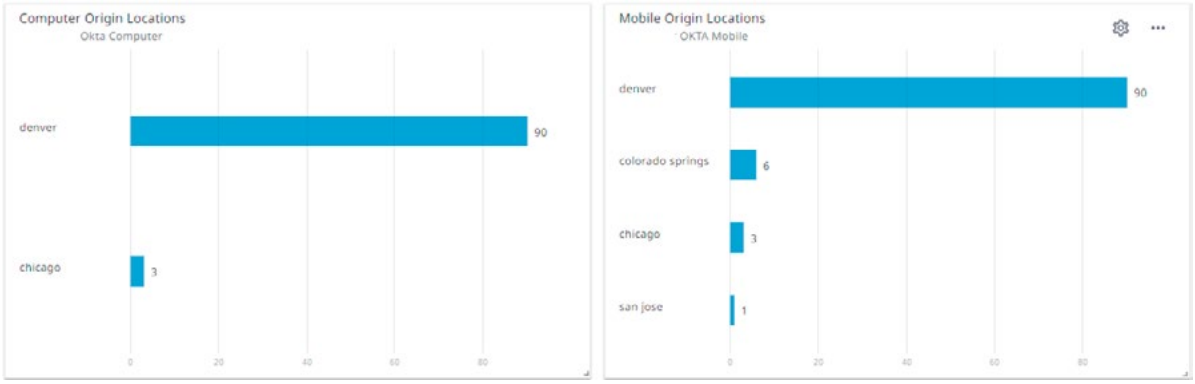
Finally, if a sophisticated and motivated attacker is involved and is attempting to compromise a high-value user, they could potentially use proxies to appear as if they are in the same location as the user they are attempting to compromise.

Exabeam Insights: Simplifying Location-Based Mismatches

It is difficult to easily automate comparing every user's authentication and push location activity; however, visualizing the associated data can provide some beneficial insight.

With New-Scale SIEM, security professionals can create custom dashboards to compare authentication and push location activity per-user easily. As shown below, the dashboard is currently filtered down to view a single user. Should that user filter be removed, a list of all users authenticating within a specific timeframe will be displayed, allowing analysts to drill into other desired user activity.

A visual analysis such as this can easily and quickly assist an SOC analyst in determining unusual user activity.



Mitigating MFA Fatigue

Safeguarding against MFA fatigue attacks can be challenging, particularly given the need to balance a streamlined authentication process while also ensuring end users are receptive and diligent in their responsiveness. While detection is as important as a reactive measure mid-attack, it is also necessary to take more proactive steps to protect users from the risk of MFA fatigue attacks. A few potential mitigation strategies include the following:

Security Policies

Given that MFA fatigue attacks begin with compromised credentials, it is important to ensure that organizational security policies include steps to strengthen password management practices. For instance, this could follow NIST's Zero Trust Architecture as a proactive approach. Furthermore, reactive measures should be considered, such as resetting a user's password following the detection or suspicion of credential compromise. Additionally, implementing the Principle of Least Privilege (POLP) concept and killing active processes and applications can reduce a threat actor's ability to access systems and data during a compromise.

Security Awareness Training

User education is another critical strategy for reducing the risk of MFA fatigue attacks. Some key components to this include ensuring users are provided guidance and training on appropriate password selection and use, how to administer and use MFA properly, and how to identify and report unsolicited push notifications and other suspicious activity. Furthermore, simulated social engineering exercises are another means for raising awareness around the implications of MFA fatigue and other similar attacks.

Additional Technical Controls

Another consideration for thwarting MFA fatigue attacks is implementing technical controls beyond the minimal user interaction offered by simple push notifications. This might include disabling push notifications and using alternative verification methods, such as temporary passcodes, where users must enter a number code from the login screen of an associated authenticator application. Another potential option is to use Fast Identity Online (FIDO), which provides two-factor authentication via USB or NFC devices, or even FIDO2, which enables passwordless authentication using biometrics.

Fighting Through the Fatigue

Until MFA applications with simple push verification capabilities disappear, MFA fatigue attacks will continue to thrive. This attack method relies on users' dismissiveness and assumption that the repeated push notifications they receive are more likely some technical glitch than an actual attack.

An organization will be better prepared to defend against MFA fatigue attacks by fully understanding their nature and adding controls to minimize the possibility of successful compromise. Furthermore, by implementing detection methods, such as those provided by New-Scale SIEM, indicators of attack activity can be discovered the moment they occur, thus reducing the risk caused by this prevalent attack technique.