Bypassing Identity Perimeter: How ClickFix and ConsentFix Subvert Microsoft 365 Sessions in Seconds

Author: Synthex **Site:** denizhalil.com **Date:** July 2026

INTRODUCTION

As organizations fortify their digital infrastructure with robust Multi-Factor Authentication (MFA) and strict conditional access guidelines, threat actors are increasingly shifting away from hacking the systems directly. Instead, modern adversaries exploit the inherent trust mechanisms within legitimate cloud architectures. Among the most dangerous of these emerging threat vectors are the ClickFix technique and its cloud-native evolution, ConsentFix. Operating at the intersection of advanced social engineering and OAuth token manipulation, these vectors bypass traditional security layers. By effectively turning legitimate, first-party utilities against the enterprise, they exploit human behavior and native protocol redirection flaws seamlessly. This weaponization of trusted cloud environments means that security perimeters are breached without generating traditional malicious alerts. Within seconds, a routine, benign user action can result in the total compromise of a Microsoft 365 environment, nullifying standard MFA implementations entirely and granting adversaries persistent, unauthorized cloud access.

LEARNING OBJECTIVES

- Understand the fundamental differences and operational mechanisms of ClickFix and ConsentFix.
- Analyze the end-to-end tactical execution of a token-theft campaign targeting Microsoft 365.
- Identify why traditional defenses like MFA fail to intercept these authorization-level attacks.
- Implement actionable enterprise mitigation and architecture hardening strategies to reduce organizational exposure.

WHAT ARE CONSENTFIX AND CLICKFIX

The modern threat landscape is increasingly defined by shifting attack methodologies that target identity boundaries rather than network vulnerabilities. Among these, the integration of behavioral manipulation with technical protocol manipulation has yielded highly successful exploits. Threat actors now actively design deceptive environments that coerce users into unknowingly compromising their own access rights. By abusing the trust built into routine administrative fixes and cloud-native workflows, adversaries bypass standard endpoint and perimeter defenses with remarkable ease. At its core, this dual-threat framework relies on user action to execute the final stage of the compromise, shifting the burden of system breach from software vulnerabilities to human execution. While one

branch of this methodology focuses on endpoint command execution through deceptive technical errors, the other elevates the concept into the cloud architecture. Both approaches minimize the footprint of traditional indicators of compromise (IoCs), ensuring that security event logs view the activity as legitimate user-initiated maintenance or application approval.

The divergence between these techniques lies strictly in their execution environment and the specific assets they target. The foundational approach weaponizes local administrative utilities to establish a foothold on physical endpoints. Conversely, the cloud-native variant leverages the complex relationships within the OAuth 2.0 framework to directly seize active session parameters, eliminating the need to interact with the underlying operating system entirely.

- **ClickFix:** This technique represents a highly coercive form of social engineering that tricks users into executing malicious scripts under the guise of fixing a system error. Typically, a user encounters a compromised or spoofed webpage showing a fake overlay—such as a Google Chrome update failure, a OneDrive synchronicity error, or a broken CAPTCHA widget. Rather than dropping an executable, the site instructs the user to press a sequence of keyboard keys (e.g., `Windows + R`), paste a heavily obfuscated string from their clipboard, and hit enter. Unwittingly, the user runs a PowerShell command that pulls down and executes remote access trojans (RATs) or infostealers, using native operating system binaries to evade antivirus detection.

- **ConsentFix:** ConsentFix abstracts the core behavioral manipulation of ClickFix and translates it into the cloud ecosystem, specifically targeting OAuth 2.0 implementation dynamics. Unlike its predecessor, ConsentFix requires absolutely no malware execution on the target's endpoint. It leverages legitimate Microsoft authentication endpoints and abuses multi-tenant application registration architectures or trusted first-party developer utilities (such as Azure CLI, Azure PowerShell, or Microsoft Graph tools). By tricking the victim into sharing authorization tokens generated during a legitimate login phase, the attacker avoids raising alarms with endpoint detection and response (EDR) platforms.

HOW THE NEW ATTACK TARGETS MICROSOFT 365 SESSIONS

The operational cadence of a ConsentFix attack on a Microsoft 365 tenant is highly structured, compressing the time-to-compromise to a matter of seconds through a perfectly orchestrated chain of events. Unlike traditional phishing campaigns that rely on crude lookalike domains to harvest static passwords, this methodology treats the official authentication infrastructure as an ally. By maintaining the entire initial interaction within legitimate parameters, the adversary ensures that network-level defenses, secure email gateways, and endpoint detection utilities remain completely blind to the unfolding compromise. The profound danger of this vector lies in its psychological precision and deep understanding of enterprise cloud logic. Threat actors exploit the gap between user authentication and client authorization, transforming a standard browser error into an actionable trap. Because corporate users are conditioned to follow instructions to resolve technical issues, the transition from a legitimate login process to a malicious exfiltration script feels like a routine troubleshooting step rather than a security breach.

Ultimately, this workflow completely shifts the dynamic of cloud security by neutralizing the concept of identity validation. The attacker never needs to crack complex passwords, guess security questions, or intercept real-time push notifications on a mobile device. Instead, they let the authorized user do all the heavy lifting, stepping in at the final fraction of a second to siphon the cryptographic proof of the validated session and gain unhindered access.

- **Step 1: The Initial Lure & First-Party Redirection:** The attacker redirects the victim to the genuine Microsoft login portal (`login.microsoftonline.com`). Because the URL explicitly references the native client ID of a highly trusted, native Microsoft application (such as Azure CLI, Azure PowerShell, or Microsoft Graph tools), automated security gateways recognize the traffic as entirely legitimate and safe.
- **Step 2: Legitimate Authentication & MFA Fulfillment:** The user completes their standard enterprise authentication workflow by inputting their real credentials and successfully passing the organizational MFA prompt. Because this occurs entirely directly on Microsoft's official infrastructure, the MFA mechanism functions exactly as designed and validates the session, believing a legitimate application is being authorized.
- **Step 3: The Localhost Redirection & ClickFix Trap:** Upon successful authentication, native utilities like Azure CLI naturally redirect to a local loopback address (e.g., `http://localhost:8400/...`) to pass the temporary authorization code back to the local machine. Since no local service is listening, the user's browser displays a standard "Site Cannot Be Reached" error page. The malicious site framing the session immediately steps in, mimicking a technical support script: *"Connection failure detected. To resolve this sync issue, copy the full URL from your browser's address bar and paste it below."*
- **Step 4: Exfiltration & Instant Token Generation:** The moment the user pastes the URL into the malicious form, the attacker extracts the ephemeral `Authorization Code` embedded within the query string. The attacker's backend infrastructure instantly transmits this code to Microsoft's token endpoint, exchanging it for full `Access Tokens` and `Refresh Tokens` tied to the victim's profile, thereby securing unhindered access to mailboxes, OneDrive vaults, and Teams channels.

HOW TO MINIMIZE YOUR EXPOSURE RATE

Mitigating the threat of advanced identity-centric exploits requires a fundamental shift from passive user awareness to proactive, structural cloud architecture hardening. Relying solely on perimeter security or baseline authentication checks is no longer sufficient when adversaries exploit trusted protocols. Organizations must implement layered defensive controls that treat every authorization flow with continuous scrutiny, regardless of whether it originates from a trusted application or a verified identity. A comprehensive defense strategy integrates strict identity governance with automated real-time response mechanisms. By limiting the blast radius of highly permissive first-party tools and eliminating unauthorized third-party integrations, the enterprise can successfully close the operational gaps that ConsentFix and ClickFix rely on. These controls ensure that even if an employee is successfully manipulated by a technical support ruse, the underlying system blocks the execution and exfiltration attempts automatically.

Ultimately, minimizing your organization's exposure rate demands a blend of technical restrictions and advanced operational coaching. Identity security must evolve to protect not just the point of login, but the entire lifecycle of the cryptographic tokens that sustain cloud sessions. Hardening the following architectural areas forms the cornerstone of this defensive depth:

- **Implement Strict Conditional Access Policies:** Configure Microsoft Entra ID Conditional Access policies to explicitly restrict or block authentication requests for powerful first-party applications (like Azure CLI or PowerShell) unless they originate from compliant, hybrid Azure AD-joined devices or trusted corporate IP ranges.
- **Enforce App Consent Restrictions:** Transition your enterprise app consent settings away from user-driven models. Require explicit administrator workflow approvals for any application registration requests, drastically minimizing the footprint of malicious multi-tenant app integration attempts.
- **Continuous Session Revocation & Monitoring:** Deploy automated risk-based detection architectures. If a session token is suddenly utilized from an anomalous IP address or an impossible travel location shortly after generation, trigger automatic token revocation workflows via Microsoft Entra ID Protection.
- **Advanced User Behavioral Coaching:** Train administrators, developers, and high-value targets to never treat browser URL addresses as arbitrary text strings. Emphasize that copying and pasting authentication parameters, localhost paths, or execution arguments into external input fields constitutes an immediate breach of the identity boundary.

CONCLUSION

ClickFix and ConsentFix highlight a critical reality in modern cyber defense: a secure authentication architecture is only as strong as its authorization parameters. By weaponizing the natural errors of localhost redirection and the implicit trust placed in first-party Microsoft tools, attackers can effectively turn a victim's own valid credentials against them. This shift marks a sophisticated evolution where traditional security boundaries are subverted from within, utilizing legitimate frameworks to execute malicious intents. As identity parameters increasingly substitute traditional networks as the main security perimeter, organizations must respond with defensive depth. The traditional reliance on perimeter firewalls has given way to an environment where identity itself must be continuously verified and constrained. Security teams must recognize that a verified login is no longer the final victory line but rather the beginning of a continuous trust evaluation process. Relying solely on MFA is no longer sufficient; defending against token-theft vectors demands rigorous application lifecycle management, continuous session validation, and strict device-compliance governance. To survive this era of token-theft and advanced social engineering, enterprises must adopt a strict Zero Trust philosophy regarding application permissions, ensuring that no single user action can inadvertently hand over the keys to the entire cloud infrastructure.