INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-02 18:36 UTC

ConsentFix v3: Automated OAuth Phishing Campaign Bypasses MFA Against Azure Environments

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0262
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Microsoft Azure, Microsoft Entra ID, Azure CLI, Microsoft first-party FOCI apps, OAuth2 authorization flow; delivery infrastructure via Cloudflare Pages, Pipedream, DocSend
Published	2026-05-02T10:32:25
Discovery Source	Rss

Executive Summary

ConsentFix v3 is an automated phishing campaign targeting Microsoft Azure and Entra ID environments by abusing legitimate OAuth2 authorization flows against pre-trusted Microsoft first-party applications. Because the attack never asks victims for passwords and exploits trust relationships built into Microsoft's identity platform by design, MFA provides no protection. Organizations using Azure face risk of unauthorized access to cloud resources, email, and sensitive data with minimal warning to end users.

Technical Analysis

ConsentFix v3 automates the full OAuth2 authorization code grant attack chain against Microsoft Azure and Entra ID. The technique targets Family of Client IDs (FOCI)-eligible, pre-consented first-party Microsoft applications, including Azure CLI and similar trusted apps, meaning no additional consent prompt is displayed to the victim. The attack chain covers tenant reconnaissance, phishing lure delivery via Cloudflare Pages, Pipedream, and DocSend (to add legitimacy), OAuth token capture, and automated exfiltration. Because authentication completes via the OAuth token grant rather than credential submission, MFA is architecturally bypassed, it is never invoked. Relevant CWEs: CWE-287 (Improper Authentication), CWE-352 (Cross-Site Request Forgery), CWE-601 (URL Redirection to Untrusted Site). MITRE ATT&CK coverage includes T1528 (Steal Application Access Token), T1550.001 (Use Alternate Authentication Material: Application Access Token), T1566.002 (Spearphishing Link), T1078.004 (Valid Accounts: Cloud Accounts), T1556.006 (Multi-Factor Authentication Bypass), T1567 (Exfiltration Over Web Service), T1583.006 (Acquire Infrastructure: Web

Services), T1589 (Gather Victim Identity Information). No CVE has been assigned; the abused trust relationships exist by design, not misconfiguration. No vendor patch resolves this, mitigation is architectural and behavioral. Sources: Microsoft Security Blog (T1), Microsoft Tech Community (T1), BleepingComputer (T3).

Action Checklist

1. Containment, Audit Entra ID enterprise applications immediately. In the Azure portal, navigate to Entra ID > Enterprise Applications and review all OAuth app grants, paying specific attention to first-party Microsoft apps with delegated permissions. Revoke suspicious OAuth tokens via the Entra ID revoke sign-in sessions and revoke refresh tokens controls for affected accounts. Consider restricting user consent to apps from verified publishers only (Entra ID > Enterprise Applications > Consent and Permissions > User Consent Settings).
2. Detection, Query Entra ID Sign-In Logs and Audit Logs for: OAuth authorization code grants to first-party Microsoft applications (AppId matches known FOCI app list) from unfamiliar devices or locations; sign-in events with resource 'Azure CLI' or 'Microsoft Office' where the client IP is associated with Cloudflare, Pipedream, or DocSend infrastructure; AuditLogs for 'Consent to application' events not initiated by IT staff. Enable and review Entra ID Identity Protection risk detections for token anomalies. The Elastic detection rule for 'Initial Access: Identity OAuth Phishing via First-Party Microsoft Application' (detection.fyi, T3 source) provides a starting query baseline, validate against your environment before production deployment.
3. Eradication, Restrict OAuth user consent permissions: set user consent to 'Do not allow user consent' or limit to apps from verified publishers via Entra ID > Enterprise Applications > User Settings. Enable the Entra ID admin consent workflow so all OAuth app authorizations route through IT approval. Review and remove any OAuth grants made to first-party apps during the suspected exposure window using the Get-AzureADOAuth2PermissionGrant PowerShell cmdlet or the Microsoft Graph API (oauth2PermissionGrants endpoint).
4. Recovery, After revoking affected tokens and tightening consent policy, validate that no persistent access remains: check for new service principals, app registrations, or delegated permission grants created during the attack window. Monitor Entra ID sign-in logs for 48-72 hours post-remediation for re-authentication attempts using the same app IDs or source IPs. Confirm conditional access policies are enforced for all cloud app access, including compliant device requirements where applicable.
5. Post-Incident, Conduct a full OAuth app inventory and enforce least-privilege on all delegated and application permissions. This campaign exposes a structural gap: user-facing MFA does not protect against token-based authentication abuse. Evaluate Continuous Access Evaluation (CAE) in Entra ID to reduce token lifetime and increase revocation responsiveness. Review MITRE ATT&CK T1528 and T1550.001 mitigations for token theft hardening. Brief security awareness programs should address link-based OAuth consent lures delivered via legitimate hosting platforms.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to legal, privacy counsel, and executive leadership immediately if Entra ID Audit Logs confirm 'Consent to application' events with scopes including Mail.Read, Mail.ReadWrite, or Files.ReadWrite.All for accounts with access to PII, PHI, or regulated financial data, as this constitutes a data access event requiring breach notification analysis under GDPR, HIPAA, or applicable state privacy laws.
Recovery Notes	After token revocation and consent policy lockdown, monitor Entra ID Sign-In Logs and the M365 Unified Audit Log (MailItemsAccessed, FileAccessed operations) for 72 hours for any re-authentication attempts originating from the attacker-associated Cloudflare Pages, Pipedream, or DocSend IP ranges using the same FOCI app IDs. Verify Conditional Access policies block non-compliant and unmanaged devices from completing OAuth flows to Azure CLI and Microsoft Office app IDs specifically. Confirm no Exchange Online inbox rules, SharePoint webhooks, or Azure Logic App connections were created during the attack window using delegated permissions, as these can maintain attacker persistence after refresh tokens are revoked.
Forensic Artifacts	Entra ID Audit Logs — OperationName: 'Consent to application' and 'Add delegated permission grant' events; filter actorType = 'User' and targetResources.displayName matching Azure CLI (appId: 04b07795-8ddb-461a-bbee-02f9e1bf7b46) or other FOCI app IDs; these are the primary record of which accounts were successfully phished via the ConsentFix v3 OAuth redirect flow | Entra ID Sign-In Logs — filter resourceDisplayName = 'Azure CLI' or 'Microsoft Office', authenticationProtocol = 'oAuth2', clientAppUsed = 'Browser', with ipAddress correlating to Cloudflare IP ranges (available at cloudflare.com/ips) or Pipedream/DocSend egress IPs identified during the investigation; userAgent strings may reveal the Cloudflare Pages redirect page | Microsoft 365 Unified Audit Log — MailItemsAccessed and FileAccessed operations for affected accounts during and after the consent event window; these record whether the attacker exercised the delegated Mail.Read or Files.Read scopes obtained through the OAuth phishing grant before tokens were revoked | Microsoft Graph oauth2PermissionGrants API snapshot — timestamped export of all delegated permission grants (`GET /oauth2PermissionGrants`) capturing clientId, resourceId, principalId, and scope fields; this is the authoritative record of what access was granted and to which FOCI apps, used for both scope-of-breach analysis and eradication verification | Exchange Online inbox rules and SharePoint webhook registrations — query via `Get-InboxRule -Mailbox ` for rules created during the attack window (forward, redirect, or delete rules are common post-compromise persistence mechanisms using delegated Mail access) and via Graph API (`GET /subscriptions`) for webhook subscriptions that could exfiltrate data after token revocation

Per-Action IR Details

Containment — Audit Entra ID enterprise applications immediately. In the Azure portal, navigate to Entra ID > Enterprise Applications and review all OAuth app grants, paying specific attention to first-party Microsoft apps with delegated permissions. Revoke suspicious OAuth tokens via the Entra ID revoke sign-in sessions and revoke refresh tokens controls for affected accounts. Consider restricting user consent to apps from verified publishers only (Entra ID > Enterprise Applications > Consent and Permissions > User Consent Settings).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST AC-17 (Remote Access), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Without a SIEM, use PowerShell with the Microsoft Graph SDK or AzureAD module to enumerate and revoke grants at scale: `Get-AzureADOAuth2PermissionGrant -All $true | Where-Object { $_.Scope -match 'Mail|Files|User' } | Export-Csv oauth_grants.csv`. Then revoke per affected UPN with `Revoke-AzureADUserAllRefreshToken -ObjectId `. A 2-person team can split: one auditing the CSV output against a known-good FOCI app list (Azure CLI appId: 04b07795-8ddb-461a-bbee-02f9e1bf7b46, Office appId: d3590ed6-52b3-4102-aeff-aad2292ab01c), the other executing revocations for flagged accounts.

Evidence: Before revoking tokens, export the full oauth2PermissionGrant snapshot via Microsoft Graph (`GET /oauth2PermissionGrants?$top=999`) and preserve raw Entra ID Audit Logs for 'Consent to application' events (OperationName: 'Consent to application', Category: 'ApplicationManagement') timestamped during the suspected attack window. Capture Entra ID Sign-In Logs filtered on resourceDisplayName containing 'Azure CLI' or 'Microsoft Office' with clientAppUsed = 'Browser' and deviceDetail.isCompliant = false, which indicates the consent flow was completed from an attacker-controlled browser rather than a managed endpoint.

Detection — Query Entra ID Sign-In Logs and Audit Logs for: OAuth authorization code grants to first-party Microsoft applications (AppId matches known FOCI app list) from unfamiliar devices or locations; sign-in events with resource 'Azure CLI' or 'Microsoft Office' where the client IP is associated with Cloudflare, Pipedream, or DocSend infrastructure; AuditLogs for 'Consent to application' events not initiated by IT staff. Enable and review Entra ID Identity Protection risk detections for token anomalies. The Elastic detection rule for 'Initial Access: Identity OAuth Phishing via First-Party Microsoft Application' (detection.fyi, T3 source) provides a starting query baseline — validate against your environment before production deployment.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, use the Microsoft Graph PowerShell module to pull and filter logs directly: `Get-MgAuditLogSignIn -Filter "resourceDisplayName eq 'Azure CLI' and ipAddress ne "" -Top 500 | Export-Csv signin_cli.csv`. For Cloudflare/Pipedream/Pipedream IP correlation, download current Cloudflare IP ranges from https://www.cloudflare.com/ips/ and cross-reference with the exported sign-in CSV using a simple PowerShell `Where-Object` filter. For consent event hunting, query: `Get-MgAuditLogDirectoryAudit -Filter "operationType eq 'Consent' and initiatedBy/user/userPrincipalName ne ""`. Schedule this as a daily task via Windows Task Scheduler during the active investigation window.

Evidence: Preserve Entra ID Sign-In Logs (retained 30 days in Entra ID P1/P2; only 7 days for free tier — export immediately) filtering on: `authenticationProtocol = 'oAuth2'`, `resourceAppId` matching known FOCI app IDs, and `conditionalAccessStatus = 'notApplied'`. Capture Entra ID Audit Logs for OperationName 'Add delegated permission grant' and 'Consent to application' with actorType = 'User' (not 'ServicePrincipal' or admin), which indicates a victim-initiated consent rather than IT-authorized grant. Flag sign-in events where `userAgent` contains browser strings consistent with a redirect page hosted on Cloudflare Pages (*.pages.dev domains) or Pipedream (pipedream.net) visible in the HTTP referrer chain if available in your WAF or proxy logs.

Eradication — Restrict OAuth user consent permissions: set user consent to 'Do not allow user consent' or limit to apps from verified publishers via Entra ID > Enterprise Applications > User Settings. Enable the Entra ID admin consent workflow so all OAuth app authorizations route through IT approval. Review and remove any OAuth grants made to first-party apps during the suspected exposure window using the Get-AzureADOAuth2PermissionGrant PowerShell cmdlet or the Microsoft Graph API (oauth2PermissionGrants endpoint).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication and Recovery

Controls: NIST IR-4 (Incident Handling), NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), NIST CM-7 (Least Functionality), NIST SI-2 (Flaw Remediation), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 6.1 (Establish an Access Granting Process)

Compensating: A 2-person team without an IGA platform can execute eradication via PowerShell in two parallel workstreams. Person 1 removes malicious grants: `$grants = Get-AzureADOAuth2PermissionGrant -All $true | Where-Object { $_.StartTime -gt (Get-Date).AddDays(-30) }; $grants | ForEach-Object { Remove-AzureADOAuth2PermissionGrant -ObjectId $_.ObjectId }`. Person 2 disables user consent via the Entra ID portal (Enterprise Applications > User Settings > 'Users can consent to apps accessing company data on their behalf' = No) and enables the admin consent workflow under the same blade. Verify the consent policy change propagated by attempting a test OAuth consent flow from a non-admin test account — it should now trigger an approval request rather than completing silently.

Evidence: Before removing grants, capture a timestamped export of all oauth2PermissionGrants created during the attack window via Graph API (`GET /oauth2PermissionGrants` filtered by `startTime`) and preserve the associated servicePrincipal objects (`GET /servicePrincipals?$filter=createdDateTime gt `). This documents which FOCI app IDs were exploited, which user accounts consented, and what permission scopes (e.g., `Mail.Read`, `Files.ReadWrite`, `User.Read`) were granted — critical for breach scope determination and any required regulatory notification.

Recovery — After revoking affected tokens and tightening consent policy, validate that no persistent access remains: check for new service principals, app registrations, or delegated permission grants created during the attack window. Monitor Entra ID sign-in logs for 48-72 hours post-remediation for re-authentication attempts using the same app IDs or source IPs. Confirm conditional access policies are enforced for all cloud app access, including compliant device requirements where applicable.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST CA-7 (Continuous Monitoring), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.5 (Require MFA for Administrative Access)

Compensating: Without a SIEM for continuous monitoring, configure Entra ID Diagnostic Settings to stream Sign-In and Audit logs to a Log Analytics workspace (free 5GB/month tier) and create a manual alert query: `SigninLogs | where AppId in ('') and IPAddress in ('')`. Alternatively, run the Graph PowerShell sign-in query as a scheduled daily task for 72 hours post-remediation and diff output against a baseline of known-good sign-ins. For service principal persistence checks, run: `Get-AzureADServicePrincipal -All $true | Where-Object { $_.CreatedDateTime -gt '' } | Select DisplayName, AppId, CreatedDateTime`.

Evidence: Before declaring recovery complete, pull a current snapshot of all servicePrincipal objects and appRoleAssignment records and diff against a pre-incident baseline (or against objects created after the earliest confirmed consent event). Specifically, look for service principals associated with FOCI app IDs (Azure CLI, Microsoft Office, Microsoft Teams) that have `oauth2PermissionGrants` with scopes beyond `User.Read` — these indicate the attacker obtained broader delegated access and may have established downstream persistence via mailbox rules, SharePoint webhooks, or Azure Automation runbooks that survive token revocation.

Post-Incident — Conduct a full OAuth app inventory and enforce least-privilege on all delegated and application permissions. This campaign exposes a structural gap: user-facing MFA does not protect against token-based authentication abuse. Evaluate Continuous Access Evaluation (CAE) in Entra ID to reduce token lifetime and increase revocation responsiveness. Review MITRE ATT&CK T1528 and T1550.001 mitigations for token theft hardening. Brief security awareness programs should address link-based OAuth consent lures delivered via legitimate hosting platforms.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST AC-6 (Least Privilege), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without a GRC platform, conduct the OAuth app inventory using Microsoft Graph PowerShell: `Get-MgServicePrincipal -All | Select DisplayName, AppId, @{N='Scopes';E={ (Get-MgServicePrincipalOauth2PermissionGrant -ServicePrincipalId $_.Id).Scope }} | Export-Csv`

oauth_inventory.csv`. Triage results by scope severity — flag any grant containing `Mail.ReadWrite`, `Files.ReadWrite.All`, `User.ReadWrite.All`, or `Directory.ReadWrite.All` for immediate review. For CAE evaluation, enable it in Entra ID under Protection > Conditional Access > Continuous Access Evaluation at no additional license cost for most tenants. Document the MFA bypass mechanism in the lessons-learned report explicitly, as this is a common misconception that MFA fully protects cloud identity — this incident is evidence for internal budget justification for Entra ID Identity Protection P2.

Evidence: Preserve the full incident timeline document including: the initial consent event timestamps from Audit Logs, the specific FOCI app IDs exploited, the OAuth scopes granted per affected account, the delivery infrastructure domains (Cloudflare Pages, Pipedream, DocSend) captured from proxy/DNS logs or browser history on victim endpoints, and any downstream actions taken with the delegated tokens (Exchange Online message access events in M365 Unified Audit Log under MailItemsAccessed operation, SharePoint/OneDrive file access events under FileAccessed). This evidence package supports both regulatory breach notification analysis (scope of data accessed) and future threat hunting for T1528 (Steal Application Access Token) and T1550.001 (Use Alternate Authentication Material: Application Access Token) recurrence.

Detection Guidance

Detection relies on behavioral signals in Entra ID audit and sign-in logs, there is no authentication-layer indicator because the OAuth flow completes legitimately from Microsoft's perspective. Key detection signals: (1) Sign-In Logs, filter for OAuth token grants where the application is a first-party Microsoft app (Azure CLI AppId: 04b07795-8ddb-461a-bbee-02f9e1bf7b46; cross-reference full FOCI app list via Microsoft documentation) and the sign-in originates from an IP not associated with the user's normal behavior or corporate egress. (2) Audit Logs, 'Consent to application' events where the initiating user is not an IT admin and the consented app is a first-party Microsoft application. (3) Unusual token usage patterns, access token use from a new device or location immediately after an OAuth authorization event. (4) Infrastructure signals, source IPs or referrer URLs associated with Cloudflare Pages (pages.dev domains), Pipedream (pipedream.net), or DocSend (docsend.com) appearing in sign-in logs as the origin of OAuth redirect completions. (5) Elastic detection rule for T1566/T1528 OAuth phishing via first-party Microsoft apps is available at detection.fyi (T3 source, validate before deployment). Microsoft Sentinel and Defender for Cloud Apps can be configured to alert on anomalous OAuth consent grants and token issuance patterns. No single indicator is definitive; correlate across sources.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`*.pages.dev`	Cloudflare Pages domains used for phishing lure delivery in ConsentFix v3 campaign infrastructure	MEDIUM
DOMAIN	`*.pipedream.net`	Pipedream automation infrastructure used for OAuth token capture and relay in ConsentFix v3	MEDIUM
DOMAIN	`docsend.com`	DocSend used to add legitimacy to phishing lure delivery chain	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1550.001** — Application Access Token
- **T1589** — Gather Victim Identity Information
- **T1556.006** — Multi-Factor Authentication
- **T1567** — Exfiltration Over Web Service
- **T1528** — Steal Application Access Token
- **T1078** — Valid Accounts
- **T1078.004** — Cloud Accounts
- **T1566.002** — Spearphishing Link
- **T1566** — Phishing
- **T1583.006** — Web Services

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **CA-7** — Continuous Monitoring
- **SC-23** — Session Authenticity
- **SI-10** — Information Input Validation
- **IA-8** — Identification and Authentication (Non-Organizational Users)

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1550.001	Application Access Token	Defense-Evasion
T1589	Gather Victim Identity Information	Reconnaissance
T1556.006	Multi-Factor Authentication	Credential-Access
T1567	Exfiltration Over Web Service	Exfiltration
T1528	Steal Application Access Token	Credential-Access
T1078	Valid Accounts	Defense-Evasion
T1078.004	Cloud Accounts	Defense-Evasion
T1566.002	Spearphishing Link	Initial-Access
T1566	Phishing	Initial-Access
T1583.006	Web Services	Resource-Development

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/consentfix-v3-attack...	T3
OAuth redirection abuse enables phishing and malware delivery	https://www.microsoft.com/en-us/security/blog/2026/03/02/oauth-redi...	T1

Source	URL	Tier
How attackers exploit Azure's elasticity for stealth and scale	https://techcommunity.microsoft.com/blog/microsoftsecurityexperts/c...	T1
How Attackers Exploit OAuth in Microsoft Entra ID to Stay in Your ...	https://medium.com/@cipherx9fsec/how-attackers-exploit-oauth-in-mic...	T3
M365 Identity OAuth Phishing via First-Party Microsoft Application	https://detection.fyi/elastic/detection-rules/integrations/o365/ini...	T3

Generated 2026-05-02 18:36 UTC by TJS Security Command Center