INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-04 18:19 UTC

Device Code Phishing Goes Mainstream: 37x Surge Signals PhaaS Maturity Threatening Identity Infrastructure

THREAT CAMPAIGN | **HIGH** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0148
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	9.5
Affected Products	Microsoft 365, Microsoft Entra ID, Microsoft Teams, SharePoint, Adobe document services, DocuSign, Citrix ShareFile, Okta
Published	2026-04-04T10:17:38
Discovery Source	Rss

Executive Summary

A device code phishing technique that bypasses multi-factor authentication has surged 37x in early 2026, driven by at least 11 commercially available phishing-as-a-service kits. Organizations running Microsoft 365, Entra ID, Teams, and SharePoint are the primary targets; successful attacks yield persistent OAuth tokens that survive password resets and credential rotation. The business risk is account takeover at scale with no MFA protection and no automatic remediation path through standard credential hygiene.

Technical Analysis

This campaign exploits the OAuth 2.0 Device Authorization Grant flow (RFC 8628). Attackers initiate a device authorization request to Microsoft's identity platform, then social-engineer the target user into visiting the legitimate device login page (login.microsoftonline.com/common/oauth2/deviceauth) and entering the attacker-supplied user code. On successful entry, the attacker receives a valid access token and refresh token bound to the user's session and scopes. The attack satisfies MFA challenges because the user completes authentication on a legitimate Microsoft endpoint; the token is issued to an attacker-controlled application registered in the tenant. Tokens persist beyond password resets and survive credential rotation unless explicitly revoked. The EvilTokens PhaaS platform is identified as a prominent kit enabling commodity-level deployment of this technique. Affected CWEs: CWE-287 (Improper Authentication), CWE-306 (Missing Authentication for Critical Function), CWE-384 (Session Fixation). Relevant MITRE ATT&CK techniques: T1528 (Steal Application Access Token), T1550.001 (Use Alternate Authentication Material: Application Access Token), T1078.004 (Valid

Accounts: Cloud Accounts), T1539 (Steal Web Session Cookie), T1566 / T1566.002 (Phishing / Spearphishing Link), T1111 (Multi-Factor Authentication Interception), T1078 (Valid Accounts). No CVE is assigned; this is an abuse of a legitimate protocol flow, not a software vulnerability. No vendor patch resolves the attack vector; mitigation requires policy and configuration controls. Source: BleepingComputer reporting on 2026 surge; technique previously documented in Microsoft threat intelligence reporting on Midnight Blizzard activity.

Action Checklist

1. Step 1: Containment. Restrict or disable the OAuth 2.0 Device Authorization Grant flow for your tenant immediately if device code authentication is not operationally required. In Microsoft Entra ID, use Conditional Access policies to block the 'Device Code Flow' authentication flow (Authentication flows condition, available in Entra ID P1/P2). Apply this policy to all users and cloud apps by default, with exceptions only for confirmed headless device use cases such as Azure CLI on headless systems.
2. Step 2: Detection. Query Microsoft Entra ID sign-in logs for authentication events where the authentication protocol field equals 'deviceCode'. In Microsoft Sentinel or Log Analytics, filter the SigninLogs table: SigninLogs | where AuthenticationProtocol == 'deviceCode' | project TimeGenerated, UserPrincipalName, AppDisplayName, IPAddress, Location, ConditionalAccessStatus, ResultType. Flag any device code authentications from users who do not administer input-constrained devices. Also review Unified Audit Log in Microsoft Purview for OAuth consent grants (Operation: 'Add delegated permission grant' or 'Consent to application') correlated with device code sign-in events within the same session window.
3. Step 3: Eradication. For any confirmed compromised accounts: (1) Revoke all active refresh tokens immediately using Microsoft Entra ID PowerShell: Revoke-MgUserSignInSession -UserId or via the Entra admin portal under the user's Authentication methods. (2) Review and revoke OAuth application consent grants made by affected users in Entra ID > Enterprise Applications > [app] > Permissions. (3) Rotate any secrets, API keys, or service credentials accessible via the compromised session. Password reset alone does not invalidate existing tokens; token revocation is the required remediation step.
4. Step 4: Recovery. After token revocation, confirm the user can re-authenticate cleanly through approved flows with MFA enforced. Verify the Conditional Access policy blocking device code flow is active and logging expected blocks in the Entra ID sign-in logs (ResultType should reflect CA block, not a successful authentication). Monitor the affected accounts for 30 days post-remediation for anomalous access patterns including unusual application access, mailbox rule creation, SharePoint access from new IP ranges, or Teams message exfiltration indicators.
5. Step 5: Post-Incident. This campaign exposes three control gaps: absence of Conditional Access restrictions on legacy and non-interactive authentication flows; insufficient OAuth application consent governance (no admin consent workflow or application allowlist); and lack of detection coverage for device code authentication events in SIEM. Address each: enforce Conditional Access authentication flow restrictions tenant-wide, implement admin consent requirements for all third-party OAuth applications, and create persistent detection rules in your SIEM for device code authentication anomalies. Review your identity threat detection coverage against T1528 and T1550.001 in MITRE ATT&CK and document gaps.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to CISO and legal counsel immediately if any compromised OAuth token held Mail.Read, Files.ReadWrite.All, or equivalent scopes on accounts with access to PII, PHI, or PCI data — the combination of silent persistent access and data exfiltration capability triggers breach notification assessment under GDPR Article 33 (72-hour window), HIPAA Breach Notification Rule, and applicable US state privacy laws; also escalate if more than 5 accounts are confirmed compromised, indicating a targeted campaign against your tenant rather than an opportunistic hit.
Recovery Notes	Token revocation via Revoke-MgUserSignInSession invalidates existing refresh tokens but does not prevent re-issuance if the device code flow Conditional Access block is not confirmed active — verify the CA policy is in Report-Only versus Enforced mode before declaring containment complete. Monitor recovered accounts for 30 days specifically for the post-compromise persistence indicators this campaign is known to leverage: inbox forwarding rules created via Mail API (MITRE T1114.003), new OAuth app consent grants that may indicate the attacker re-phished the same user, and SharePoint or Teams access originating from IP addresses outside the user's established geolocation baseline. Any recurrence of deviceCode authentication events on a remediated account within the monitoring window indicates the attacker retained a token that was not captured in the revocation sweep and should trigger re-eradication.
Forensic Artifacts	Entra ID SigninLogs — AuthenticationProtocol = 'deviceCode' entries: these are the primary evidence of the PhaaS kit's token request interception, containing the exact timestamp, user, application (often spoofed as Microsoft Office or Adobe), attacker IP, and ConditionalAccessStatus at time of token issuance; retain for minimum 90 days | Microsoft Purview Unified Audit Log — 'Consent to application' and 'Add delegated permission grant' operations: these record the exact Microsoft Graph API scopes delegated to the attacker-controlled application at the moment the victim completed the device code flow, defining the full data access capability of the stolen token | Entra ID Oauth2PermissionGrants export (Get-MgOauth2PermissionGrant): point-in-time snapshot of all active delegated permission grants, captured before eradication, documenting which applications held which Graph API scopes on which user accounts — this is the definitive record of the token's blast radius | Exchange Online mailbox rule audit (Get-InboxRule per affected mailbox): attacker-planted inbox forwarding or deletion rules created via Mail API after token issuance are a known post-compromise persistence mechanism in Microsoft 365 account takeover campaigns; rules with ForwardTo, ForwardAsAttachmentTo, or DeleteMessage set after the device code authentication timestamp are high-confidence indicators of active data exfiltration | Entra ID Audit Log — 'Update application' and 'Add service principal' operations correlated with the attack window: PhaaS kits targeting Microsoft 365 and Entra ID frequently register rogue service principals or modify existing app registrations to establish durable access; these operations appear in the Entra audit log and must be reviewed to confirm no persistent application-level backdoor was planted beyond the user-level OAuth grant

Per-Action IR Details

Step 1: Containment — Restrict or disable the OAuth 2.0 Device Authorization Grant flow for your tenant immediately if device code authentication is not operationally required. In Microsoft Entra ID, use Conditional Access policies to block the 'Device Code Flow' authentication flow (Authentication flows condition, available in Entra ID P1/P2). Apply this to all users and all cloud apps as the default deny posture, with explicit exceptions only for confirmed input-constrained device use cases such as Azure CLI on headless systems.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: stopping the spread of the incident by isolating the attack vector; here the vector is the Device Authorization Grant flow itself, not an endpoint or network segment

Controls: NIST IR-4 (Incident Handling) — execute the containment phase of the incident handling capability, NIST AC-3 (Access Enforcement) — enforce approved authorizations for device code flow at the identity-plane boundary via Conditional Access, NIST SC-8 (Transmission Confidentiality and Integrity) — restrict authentication flows to only those channels that enforce MFA and session integrity, CIS 6.3 (Require MFA for Externally-Exposed Applications) — Conditional Access policy blocking device code flow is the enforcement mechanism for this safeguard against non-MFA-capable OAuth flows, CIS 4.4 (Implement and Manage a Firewall on Servers) — applied in the identity plane: default-deny on authentication flows with explicit allow-list exceptions mirrors firewall posture

Compensating: Without Entra ID P1/P2, use PowerShell to enumerate and disable device code flow at the application registration level: Connect-MgGraph -Scopes 'Application.ReadWrite.All'; Get-MgApplication | Where-Object { $_.PublicClient -ne $null } | Select-Object DisplayName, AppId. For each non-essential public client app identified, set the fallback restriction via: Update-MgApplication -ApplicationId -PublicClient @{RedirectUris=@()}. Additionally, use the free Microsoft Entra audit logs (available without P1) to manually review the 'Authentication flows' report in the Entra portal under Identity > Monitoring for any device code sessions within the last 30 days before applying restrictions.

Evidence: Before applying the Conditional Access block, export the full Entra ID SigninLogs for the prior 90 days filtered on AuthenticationProtocol == 'deviceCode' to preserve evidence of all sessions that used this flow — including UserPrincipalName, AppDisplayName, IPAddress, DeviceDetail, and ConditionalAccessStatus fields. This establishes the pre-containment blast radius. Also capture a snapshot of all Enterprise Application consent grants via: Get-MgOauth2PermissionGrant -All | Export-Csv oauth_grants_precontainment.csv — this records which delegated permissions were active before revocation and preserves chain-of-custody evidence of OAuth token grants issued during the attack window.

Step 2: Detection — Query Microsoft Entra ID sign-in logs for authentication events where the authentication protocol field equals 'deviceCode'. In Microsoft Sentinel or Log Analytics, filter the SigninLogs table: SigninLogs | where AuthenticationProtocol == 'deviceCode' | project TimeGenerated, UserPrincipalName, AppDisplayName, IPAddress, Location, ConditionalAccessStatus, ResultType. Flag any device code authentications from users who do not administer input-constrained devices. Also review Unified Audit Log in Microsoft Purview for OAuth consent grants (Operation: 'Add delegated permission grant' or 'Consent to application') correlated with device code sign-in events within the same session window.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlating sign-in telemetry with OAuth consent events to determine scope of token issuance and confirm incident criteria are met for device code phishing

Controls: NIST IR-5 (Incident Monitoring) — track and document all device code authentication events and correlated OAuth consent grants across the incident timeline, NIST AU-2 (Event Logging) — ensure deviceCode authentication events and OAuth consent operations are included in the organization's defined event logging set, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — review and analyze Entra ID SigninLogs and Unified Audit Log at the frequency required to detect active device code phishing campaigns, NIST SI-4 (System Monitoring) — monitor identity infrastructure for device code authentication anomalies as indicators of this specific PhaaS campaign pattern, CIS 8.2 (Collect Audit Logs) — ensure Entra ID sign-in logs and Microsoft Purview Unified Audit Log are actively collected and retained for the investigation window

Compensating: Without Sentinel, use the free Microsoft Entra ID sign-in log export via PowerShell: Connect-MgGraph -Scopes 'AuditLog.Read.All'; Get-MgAuditLogSignIn -Filter "authenticationProtocol eq 'deviceCode'" -All | Select-Object CreatedDateTime, UserPrincipalName, AppDisplayName, IpAddress, Location, ConditionalAccessStatus, Status | Export-Csv devicecode_signins.csv. For Unified Audit Log without Purview premium, use the Exchange Online PowerShell module: Search-UnifiedAuditLog -StartDate (Get-Date).AddDays(-90) -EndDate (Get-Date) -Operations 'Add delegated permission grant','Consent to application' | Export-Csv oauth_consent_events.csv. Correlate the two CSVs manually in Excel or with a Python pandas join on UserPrincipalName and timestamp window (±15 minutes). Use the free Sigma rule 'azure_ad_device_code_phishing.yml' from the SigmaHQ repository as a detection reference when building manual queries.

Evidence: The attack leaves two correlated evidence trails: (1) Entra ID SigninLogs entries with AuthenticationProtocol = 'deviceCode', AppDisplayName matching the PhaaS kit's spoofed application name (commonly impersonating

Microsoft Office, Adobe, DocuSign, or Citrix ShareFile), and IPAddress resolving to attacker-controlled infrastructure — often residential proxies or cloud exit nodes inconsistent with the victim's normal geolocation. (2) Unified Audit Log entries for 'Consent to application' or 'Add delegated permission grant' operations time-correlated within the same session window, showing the specific Microsoft Graph API scopes delegated (e.g., Mail.Read, Files.ReadWrite.All, Calendars.Read) — these scope grants reveal the attacker's intended post-compromise access objectives and must be preserved as forensic evidence of the token's capability at time of issuance.

Step 3: Eradication — For any confirmed compromised accounts: (1) Revoke all active refresh tokens immediately using Microsoft Entra ID PowerShell: Revoke-MgUserSignInSession -UserId or via the Entra admin portal under the user's Authentication methods. (2) Review and revoke OAuth application consent grants made by affected users in Entra ID > Enterprise Applications > [app] > Permissions. (3) Rotate any secrets, API keys, or service credentials accessible via the compromised session. Password reset alone does not invalidate existing tokens — token revocation is the required remediation step.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: removing the threat from the environment; for device code phishing the threat persists as live OAuth refresh tokens that survive credential rotation, requiring explicit token revocation as the eradication action

Controls: NIST IR-4 (Incident Handling) — execute the eradication phase, specifically targeting the persistence mechanism unique to this attack: long-lived OAuth refresh tokens, NIST AC-2 (Account Management) — revoke active sessions and delegated permissions for compromised accounts as part of account remediation, NIST IA-5 (Authenticator Management) — rotate all credentials and secrets accessible via the compromised OAuth session scope; note that IA-5 password rotation is insufficient alone when refresh tokens remain valid, NIST SI-2 (Flaw Remediation) — remove the persistent access mechanism (OAuth grant) introduced by the phishing attack before declaring eradication complete, CIS 5.3 (Disable Dormant Accounts) — review whether any dormant accounts received device code phishing attempts and disable them to prevent token issuance on low-visibility accounts, CIS 6.2 (Establish an Access Revoking Process) — execute the documented access revocation process for OAuth grants and session tokens, not just credential revocation

Compensating: For teams without a scripted remediation workflow, execute token revocation in bulk using: $users = Import-Csv compromised_users.csv; foreach ($user in $users) { Revoke-MgUserSignInSession -UserId $user.UPN; Write-Output "Revoked: $($user.UPN)" }. For OAuth grant revocation without a Purview premium license, enumerate and remove grants via: Get-MgUserOauth2PermissionGrant -UserId | ForEach-Object { Remove-MgOauth2PermissionGrant -OAuth2PermissionGrantId $_.Id }. Document each revocation action with timestamp and operator identity to maintain chain of custody. For service account credentials exposed via compromised sessions (e.g., SharePoint app registrations, Okta API tokens), treat each as fully compromised and rotate using the respective vendor's credential rotation procedures — do not assume read-only scope grants were benign, as Mail.Read and Files.ReadWrite.All are sufficient for data exfiltration.

Evidence: Before revoking tokens, capture the full OAuth permission grant record for each compromised account: Get-MgUserOauth2PermissionGrant -UserId | Select-Object ClientId, ConsentType, PrincipalId, ResourceId, Scope | Export-Csv user_oauth_grants__prerevocation.csv. This preserves the exact Graph API scopes delegated (Mail.Read, Files.ReadWrite.All, Team.ReadBasic.All, etc.) which define the data access window during compromise and is required for breach scope assessment. Also capture MgUserMemberOf output for each compromised account to determine group memberships and SharePoint sites accessible via the stolen token — this defines the maximum data exposure boundary for any subsequent regulatory notification analysis.

Step 4: Recovery — After token revocation, confirm the user can re-authenticate cleanly through approved flows with MFA enforced. Verify the Conditional Access policy blocking device code flow is active and logging expected blocks in the Entra ID sign-in logs (ResultType should reflect CA block, not a successful authentication). Monitor the affected accounts for 30 days post-remediation for anomalous access patterns including unusual application access, mailbox rule creation, SharePoint access from new IP ranges, or Teams message exfiltration indicators.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restoring systems to normal operation with verification that the threat has been fully removed and that monitoring confirms no re-compromise via residual tokens or re-issued device code grants

Controls: NIST IR-4 (Incident Handling) — execute the recovery phase with verification steps specific to OAuth token-based persistence, NIST CA-7 (Continuous Monitoring) — maintain enhanced monitoring on recovered accounts for 30 days to detect re-compromise attempts or residual token activity from tokens not captured in the revocation sweep, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — review Entra ID sign-in logs post-revocation to confirm CA policy is blocking device code flow attempts and no successful device code authentications have occurred, NIST SI-4 (System Monitoring) — monitor for post-recovery indicators specific to this attack: mailbox rule creation (T1114.003), SharePoint access from new geographies, and anomalous Teams API activity, CIS 7.2 (Establish and Maintain a Remediation Process) — verify remediation completeness against documented criteria before closing the incident

Compensating: Without a SIEM for 30-day monitoring, implement scheduled PowerShell scripts via Windows Task Scheduler or cron: (1) Daily Entra sign-in anomaly check: Get-MgAuditLogSignIn -Filter "userId eq '' and createdDateTime ge '' | Where-Object { $_.IpAddress -notin $knownIPList } | Export-Csv daily_anomaly_check.csv. (2) Weekly mailbox rule audit via Exchange Online PowerShell: Get-InboxRule -Mailbox | Where-Object { $_.ForwardTo -ne $null -or $_.DeleteMessage -eq $true -or $_.ForwardAsAttachmentTo -ne $null } — any rules created after the device code phishing event date that forward or delete mail should be treated as attacker-planted persistence. (3) SharePoint access from new IP ranges: Search-UnifiedAuditLog -Operations FileAccessed -UserIds -StartDate | Where-Object { $_.ClientIP -notin $knownIPList } | Export-Csv sharepoint_postrecovery.csv.

Evidence: Post-revocation, capture a clean-state baseline of: (1) all active Entra ID sessions for affected users confirming zero active refresh tokens (Get-MgUserAuthenticationMethod and session state via the Entra portal); (2) current inbox rules for each affected mailbox as of recovery date to establish baseline for subsequent anomaly detection; (3) a SharePoint access report for the 30 days preceding compromise to establish normal access patterns against which post-recovery anomalies will be measured. These baselines are required to distinguish attacker-planted persistence (inbox forwarding rules, rogue OAuth apps re-consented) from normal user behavior during the monitoring window.

Step 5: Post-Incident — This campaign exposes three control gaps: absence of Conditional Access restrictions on legacy and non-interactive authentication flows; insufficient OAuth application consent governance (no admin consent workflow or application allowlist); and lack of detection coverage for device code authentication events in SIEM. Address each: enforce Conditional Access authentication flow restrictions tenant-wide, implement admin consent requirements for all third-party OAuth applications, and create persistent detection rules in your SIEM for device code authentication anomalies. Review your identity threat detection coverage against T1528 and T1550.001 in MITRE ATT&CK and document gaps.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: lessons learned and control improvements to prevent recurrence; specifically addressing the three structural gaps this device code phishing campaign exploited in identity infrastructure

Controls: NIST IR-4 (Incident Handling) — update the incident handling capability to include device code phishing as a named scenario with detection and containment procedures, NIST IR-8 (Incident Response Plan) — revise the IR plan to document OAuth token revocation as a required remediation step distinct from password reset for identity-based incidents, NIST SI-5 (Security Alerts, Advisories, and Directives) — incorporate threat intelligence on PhaaS device code phishing kits (11 identified kits as of early 2026) into ongoing security advisory monitoring, NIST RA-3 (Risk Assessment) — document the residual risk from legacy and non-interactive authentication flows as a named risk item following this incident, NIST AU-2 (Event Logging) — add deviceCode authentication protocol events to the defined event logging set and verify SIEM ingestion of Entra ID SigninLogs at the required retention period, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — incorporate identity authentication flow restrictions as a recurring configuration review item, CIS 6.3 (Require MFA for Externally-Exposed Applications) — close the identified gap: admin consent workflow and application allowlist enforcement for third-party OAuth apps are required to satisfy this safeguard, CIS 8.2 (Collect Audit Logs) — verify Unified Audit Log collection includes OAuth consent operations and is retained for minimum 90 days to support future device code phishing investigations

Compensating: For teams without enterprise SIEM, implement persistent detection using: (1) A free Sigma rule for device code phishing detection — translate the following logic to your log platform: title: 'Entra ID Device Code Authentication'; logsource: product: azure, service: signinlogs; detection: selection: AuthenticationProtocol: deviceCode; condition: selection. The SigmaHQ repository contains azure_ad_device_code_phishing.yml as a starting reference. (2) For admin consent workflow without Entra ID P2, enable tenant-level user consent restrictions via the free Entra ID portal setting: Identity > Enterprise Applications > Consent and permissions > User consent settings — set to 'Do not allow user consent' and require admin approval for all third-party app OAuth grants. (3) For ATT&CK gap mapping without a commercial tool, use the free MITRE ATT&CK Navigator (https://mitre-attack.github.io/attack-navigator/) to layer T1528 (Steal Application Access Token) and T1550.001 (Use Alternate Authentication Material: Application Access Token) against your current detection coverage and export the gap map for documentation.

Evidence: For the lessons-learned record and future threat hunting baseline, preserve: (1) the complete list of PhaaS kit application display names and client IDs observed in your tenant's device code authentication events — these serve as threat intelligence IOCs for future hunting; (2) a documented inventory of all third-party OAuth applications that held active consent grants at the time of the incident, including their Graph API permission scopes, as the basis for the post-incident application allowlist; (3) the Conditional Access policy audit log showing the before/after state of authentication flow restrictions, timestamped, to document the control gap closure for GRC purposes and any applicable regulatory reporting obligations.

Detection Guidance

Primary detection surface is Microsoft Entra ID sign-in logs. Query SigninLogs for AuthenticationProtocol == 'deviceCode'. Correlate against your known inventory of legitimate device code users (headless servers, Azure CLI pipelines, Teams Rooms devices). Any device code authentication from a standard user workstation or from a user with no registered input-constrained device is a high-confidence anomaly. Secondary signals: OAuth application consent grants appearing within minutes of a device code sign-in event; new application access patterns on the account (first-time access to SharePoint, Exchange, or Teams via a registered app); mail forwarding rules created post-authentication; access from IP geolocation inconsistent with the user's prior sign-in history. In Microsoft Sentinel, map these to MITRE T1528 and T1550.001. If using Microsoft Defender for Cloud Apps, enable anomaly detection policies for 'Impossible travel' and 'Activity from anonymous IP addresses' scoped to OAuth app sessions. Public IOCs (IPs, domains, hashes) attributed to EvilTokens have not been released in available sources. Focus detection on behavioral indicators rather than static IOCs.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`login.microsoftonline.com/common/oauth2/deviceauth`	Legitimate Microsoft device code entry endpoint — abused in this attack flow. Presence in proxy logs alone is not malicious; correlate with user context and authentication outcome.	LOW

Framework Mappings

MITRE-ATTACK

- **T1539** — Steal Web Session Cookie

- **T1566.002** — Spearphishing Link
- **T1078.004** — Cloud Accounts
- **T1528** — Steal Application Access Token
- **T1550.001** — Application Access Token
- **T1111** — Multi-Factor Authentication Interception
- **T1566** — Phishing
- **T1078** — Valid Accounts

NIST-800-53R5

- **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
- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **IA-8** — Identification and Authentication (Non-Organizational Users)

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **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

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
- **164.312(e)(1)** — Transmission Security

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1539	Steal Web Session Cookie	Credential-Access
T1566.002	Spearphishing Link	Initial-Access
T1078.004	Cloud Accounts	Defense-Evasion
T1528	Steal Application Access Token	Credential-Access
T1550.001	Application Access Token	Defense-Evasion
T1111	Multi-Factor Authentication Interception	Credential-Access
T1566	Phishing	Initial-Access
T1078	Valid Accounts	Defense-Evasion

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/device-code-phishing...	T3
Disrupting active exploitation of on-premises SharePoint ... - Microsoft	https://www.microsoft.com/en-us/security/blog/2025/07/22/disrupting...	T1
This Microsoft Entra ID Vulnerability Could Have Been Catastrophic	https://www.wired.com/story/microsoft-entra-id-vulnerability-digita...	T2
Microsoft Entra ID Vulnerability Exposes Identity Security Risks	https://checkred.com/resources/blog/microsoft-entra-id-vulnerabilit...	T3
Microsoft's Entra ID vulnerabilities could have been catastrophic	https://arstechnica.com/civis/threads/microsoft%E2%80%99s-entra-id-...	T2

Generated 2026-04-04 18:19 UTC by TJS Security Command Center