ICT DIVISION
FUTURE IS HERE

BANGLADESH COMPUTER COUNCIL
COMPUTER FOR EVERYTHING

BGD e-GOV CIRT

TLP:CLEAR

Ghost Phishing: AES-GCM-Encrypted Lures and the EvilTokens Device-Code Phishing-as-a-Service Kit Targeting Microsoft 365 Accounts

TLP: CLEAR
Distribution: Public
Advisory on: Ghost Phishing: AES-GCM-Encrypted Lures and the EvilTokens Device-Code Phishing-as-a-Service Kit Targeting Microsoft 365 Accounts
Severity: High
Threat Type: Phishing / Identity & Access Compromise (OAuth Device Code Abuse)
Date: 13 July 2026

Executive Summary

BGD e-GOV CIRT is issuing this advisory to raise awareness of EvilTokens, a Phishing-as-a-Service (PhaaS) kit that abuses the legitimate Microsoft OAuth 2.0 Device Authorization Grant (device code) flow to steal persistent Microsoft 365 access and refresh tokens without requiring password theft or triggering suspicious MFA prompts.

The campaign employs a "ghost phishing" technique in which phishing content is delivered in an AES-GCM encrypted format and decrypted only within the victim's browser. This browser-based rendering significantly reduces the visibility of the malicious content during conventional security inspection.

Recent research has shown that attackers are further enhancing this technique through AI-generated phishing lures, disposable cloud-hosted infrastructure, and dynamic code generation. Originally observed in state-sponsored operations, device-code phishing has since evolved into a commoditized attack method offered through Phishing-as-a-Service platforms, enabling a broader range of threat actors to target Microsoft 365 environments.

Although reported victims have primarily been organizations in North America and Europe, the technique can target any organization using Microsoft 365 / Entra ID. BGD e-GOV CIRT recommends that government agencies, financial institutions, educational organizations, and other Microsoft 365 users in Bangladesh implement the mitigation, detection, and monitoring measures provided in this advisory to reduce the risk of compromise.

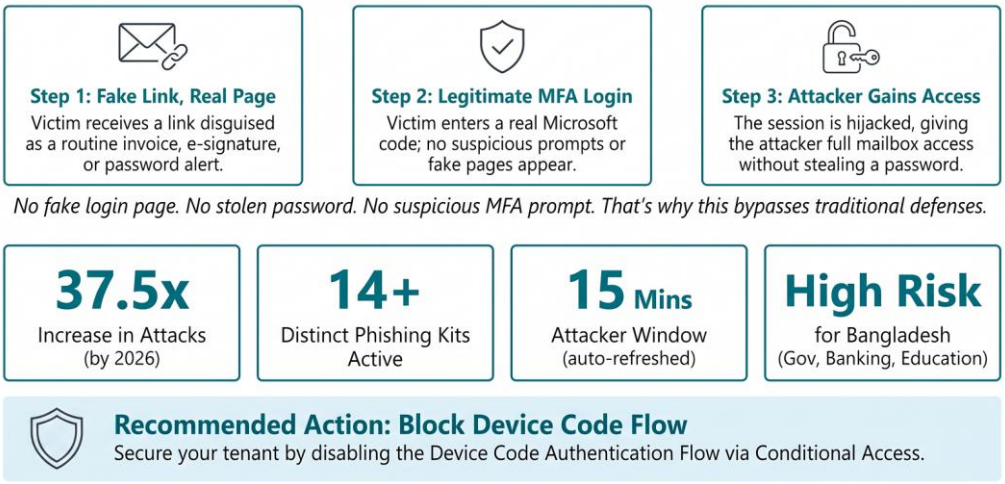

Figure: Ghost Phishing at a Glance

What Is Device Code Phishing?

The OAuth 2.0 Device Authorization Grant ("device code flow") is a legitimate authentication mechanism designed for devices with limited input capability -- smart TVs, IoT devices, CLI tools, and conference-room hardware. The device displays a short alphanumeric code and instructs the user to browse to a URL (typically microsoft.com/devicelogin) on a separate, capable device (e.g., a phone or laptop) to enter the code and complete sign-in.

The flow is exploitable because the resulting authenticated session is not cryptographically bound to the device or browser context that initiated the request. An attacker can initiate the device code request themselves, then use social engineering to get the victim to enter that attacker-generated code on the genuine Microsoft sign-in page. When the victim completes their own password and MFA challenge as normal, they unknowingly authorize the attacker's session -- and the attacker's backend receives a valid, working access token and refresh token. No password is phished, and because the victim's own MFA method was used on the genuine Microsoft page, no anomalous MFA prompt is triggered.

This makes device code phishing structurally different from classic credential-harvesting phishing (fake login page collecting a password) or even adversary-in-the-middle (AitM) proxy kits (which relay a live session through attacker infrastructure). Device code phishing needs no fake login page at all -- the victim genuinely authenticates to Microsoft; only the authorization decision is misdirected.

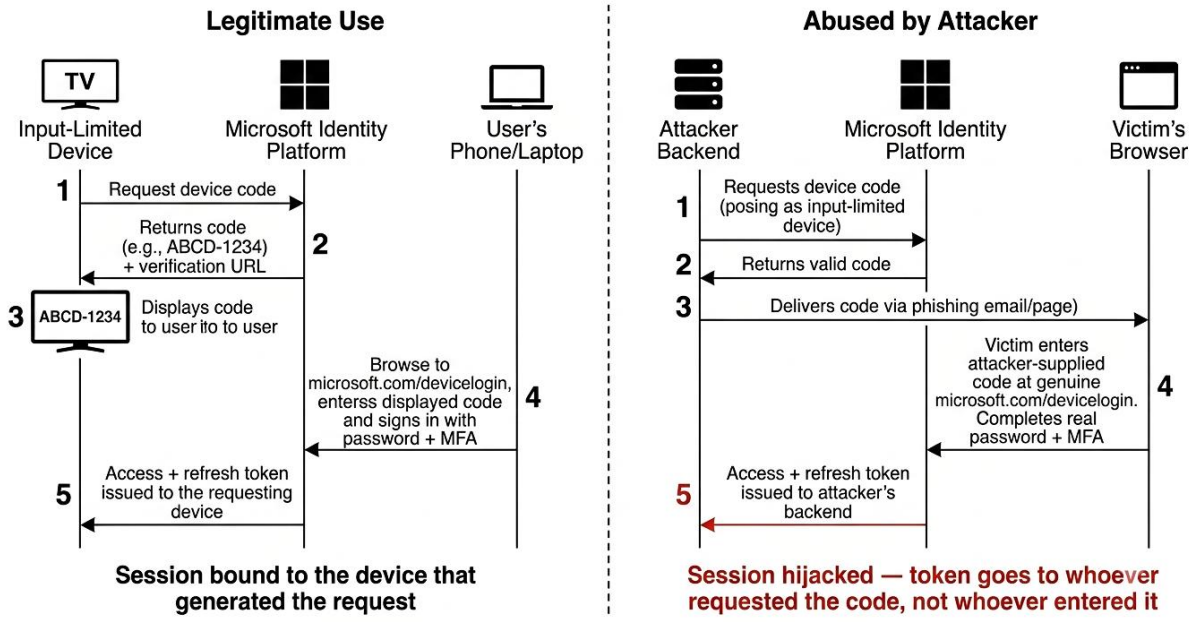

Figure: Legitimate vs. Abused Device Code Flow

Campaign Timeline and Threat Actor Evolution

Date	Development
Aug 2024	Microsoft first observes large-scale device code phishing tradecraft used by Storm-2372, an actor assessed to align with Russian state interests.
Feb 2025	Microsoft publicly documents the Storm-2372 device code phishing campaign.
Mid-Feb 2026	EvilTokens phishing pages first observed in the wild.
Mar 2026	Financially motivated actor TA4903 adopts device code phishing, per Proofpoint reporting -- marking commoditization beyond nation-state use.
Mar 2026	Huntress documents Railway.com-hosted replay/polling infrastructure used for Microsoft 365 token theft at scale.
30 Mar 2026	Sekoia formally identifies and names the EvilTokens Phishing-as-a-Service (PhaaS) kit, sold via Telegram.
6 Apr 2026	Microsoft Defender Security Research publishes findings on an AI-driven, end-to-end automated device code phishing campaign, tying it to the EvilTokens ecosystem.
2026 (year to date)	Push Security measures a 37.5-fold increase in detected device-code phishing pages, up from a 15-fold increase measured in early March 2026; 14+ distinct kits identified in the wild.
8 Jul 2026	Public reporting describes a new "ghost phishing" wave combining device code phishing with AES-GCM-encrypted, browser-rendered phishing pages designed to defeat network-layer and sandbox inspection.

Technical Analysis: The Attack Chain

Phase 1: Reconnaissance and Target Validation

- Threat actors query Microsoft's GetCredentialType endpoint to confirm a targeted email address exists and is active within a tenant, typically 10-15 days before the phishing attempt is launched.
- Target lists are refined using role-based signals to prioritize financial, executive, and administrative personas.

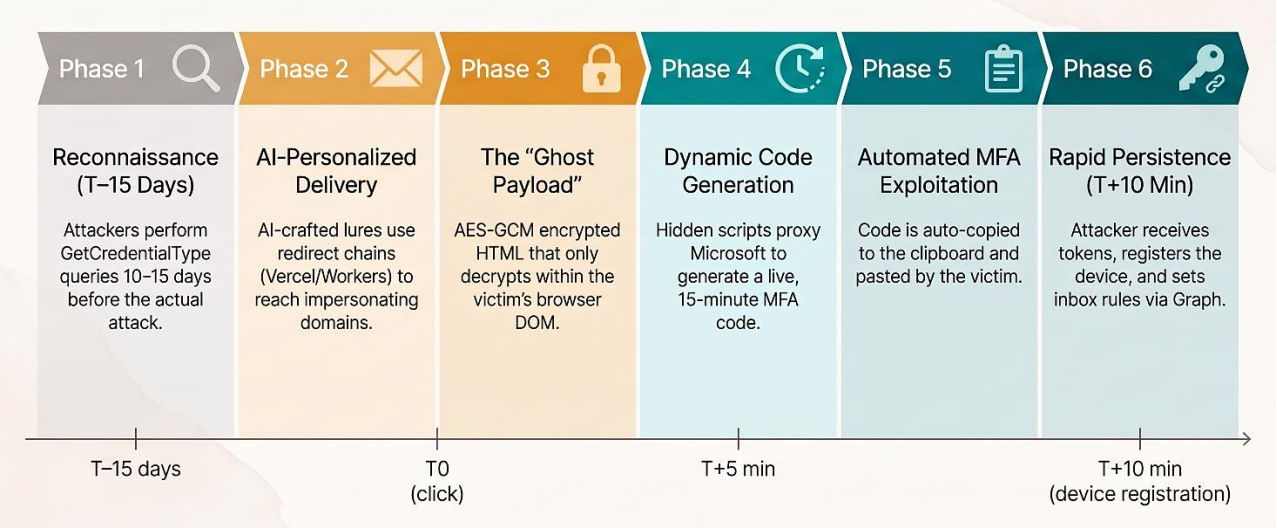

Figure: Six-Phase Attack Chain (Kill Chain Diagram)

Phase 2: Initial Access / Delivery

- Victims receive an email themed as an invoice, RFP, shared file, voicemail notification, e-signature request, or "Action Required: Password Expiration" notice. Generative AI is used to personalize lures to the victim's role and organization.
- Links do not go directly to the phishing site. Traffic is routed through redirect chains hosted on trusted "serverless" platforms *.vercel.app, *.workers.dev (Cloudflare Workers), and AWS Lambda; plus, compromised legitimate domains, to evade reputation- and blocklist-based filtering.
- Impersonating domains observed include patterns such as graph-microsoft[.]com, portal-azure[.]com, office365-login[.]com, and randomized brand-impersonating subdomains such as a7b2-c9d4.office-verify[.]net (domain shadowing).

Phase 3: The "Ghost" Layer: Encrypted, Browser-Rendered Payload

- The landing page HTML is delivered encrypted with AES-GCM. The malicious content is invisible to network-layer scanners, secure email gateways, and automated sandboxes, which see only the encrypted (harmless-looking) response.
- The page becomes malicious only after the victim's browser decrypts the payload and renders it into the live DOM -- a "ghost" that is undetectable until it materializes on the actual user's screen.
- Some landing pages present a browser-in-the-browser fake window or a blurred "document preview" to add legitimacy before prompting the "Verify identity" / "Continue to Microsoft" action.

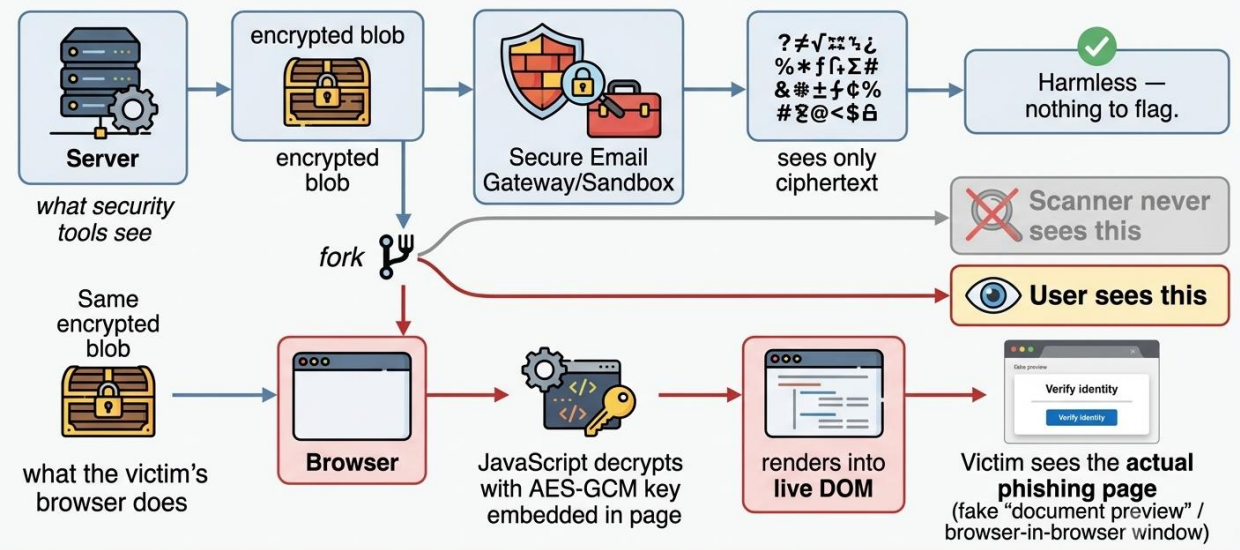

Figure: "Ghost" Payload Lifecycle; Detection implication: network/email-layer inspection is blind here — only browser/endpoint telemetry observing the rendered DOM or clipboard events can catch this phase

Phase 4: Dynamic Device Code Generation

- When the victim clicks through, a hidden automation script sends a request (empty body, content-length: 0) to the attacker backend at /api/device/start/ or /start*/, carrying a custom header X-Antibot-Token (a 64-character hex value).
- The attacker's server acts as a live proxy to Microsoft's official device authorization endpoint, requesting a fresh device code on demand and supplying the victim's email as a sign-in hint. This returns a genuine Microsoft device code with a full, un-decayed 15-minute lifespan.
- This is the critical innovation over older, static device-code phishing: earlier attacks embedded a pre-generated code directly in the email, which frequently expired before the victim acted. Dynamic, on-click generation restarts the 15-minute clock at the moment of victim interaction, dramatically raising success rates.

Phase 5: Exploitation, Clipboard Hijack, and Authentication

- The script automatically copies the generated device code to the victim's clipboard via navigator.clipboard.writeText.
- The victim is redirected to the genuine microsoft.com/devicelogin page, pastes the code, and completes their own password and MFA as normal -- unknowingly authorizing the attacker's session. If the victim already has an active browser session, confirmation alone completes the takeover.
- Because authentication runs entirely on genuine Microsoft infrastructure with the victim's own credentials and MFA method, this bypasses MFA by design rather than by defeating it technically.

Phase 6: Session Validation, Persistence, and Post-Compromise Activity

- The backend polls a /state endpoint every 3-5 seconds via a checkStatus() loop (implemented with setInterval), returning "pending" until the victim completes login, then "success" once a live access token is issued. The victim is redirected to an innocuous placeholder page (e.g., Docusign, Google, or Microsoft).
- In observed cases, attackers registered a new device within 10 minutes of compromise to obtain a Primary Refresh Token (PRT) for durable, long-term persistence independent of the original session.
- Other observed post-compromise activity: creation of malicious inbox rules for silent mail exfiltration, Microsoft Graph API reconnaissance to map organizational structure and permissions, and targeted searches for wire transfer instructions, pending invoices, and executive correspondence (i.e., business email compromise / fraud staging).
- Standard session revocation (revoking refresh tokens) can leave previously issued access tokens valid for up to an additional hour -- responders must treat this window explicitly in containment planning.

Infrastructure and Hosting Patterns

The campaign deliberately builds on reputable cloud and PaaS providers to blend with legitimate enterprise traffic and evade reputation-based network controls:

- Backend automation / polling nodes: Railway.com is used to spin up thousands of unique, short-lived Node.js backend instances end-to-end -- from dynamic device code generation through post-compromise polling.
- Additional attacker-controlled infrastructure providers: Cloudflare and DigitalOcean.
- Redirect-chain hosting (to evade URL scanners): Vercel, Cloudflare Workers, and AWS Lambda.
- Multiple legitimate, compromised third-party domains are also used to host intermediate redirect or landing content.

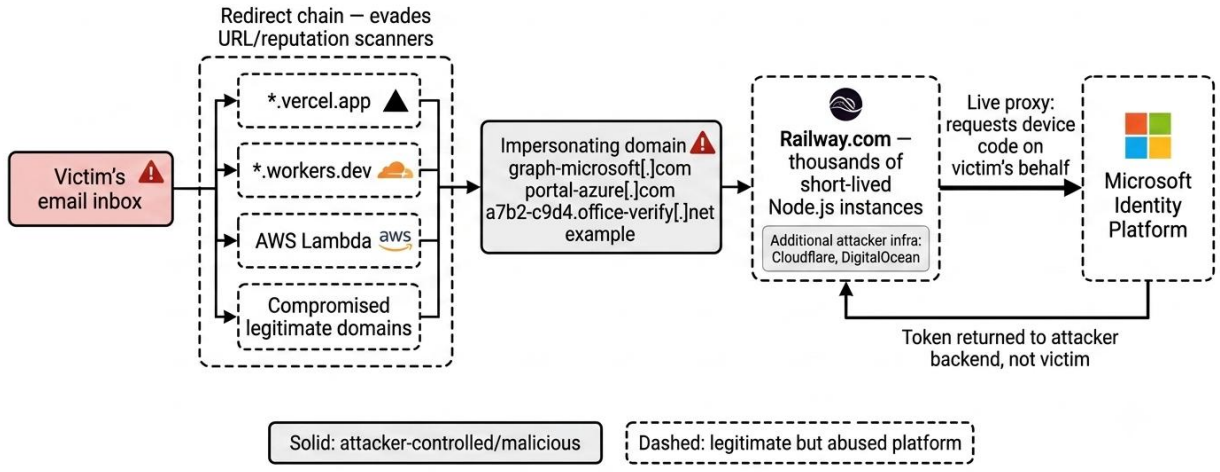

Figure: Attacker Infrastructure & Redirect-Chain Architecture

Targeted Sectors and Relevance to Bangladesh

BGD e-GOV CIRT assesses this threat as directly relevant to Bangladesh for the following reasons:

- EvilTokens is sold as a commodity Phishing-as-a-Service kit via Telegram with no regional restriction; any purchaser can target any Microsoft 365 tenant worldwide, including government, banking, and academic institutions in Bangladesh.
- The technique requires no local infrastructure compromise -- it abuses a standard Microsoft authentication flow that is enabled by default in most Entra ID / Microsoft 365 tenants.
- Bangladeshi government ministries, banks and NBFIs, universities, and private enterprises make extensive use of Microsoft 365 for email and collaboration, and would be exposed if device code flow is not explicitly restricted.
- The sector profile already targeted abroad (banking, financial services, education) directly mirrors sectors BGD e-GOV CIRT has repeatedly flagged as high-priority targets in Bangladesh in past advisories.
- The "ghost" encrypted-payload technique specifically defeats network-perimeter and email-gateway controls that many Bangladeshi organizations rely on as their primary phishing defense, increasing the practical risk of successful compromise even where basic email filtering is in place.

MITRE ATT&CK Mapping

Technique ID	Technique Name	Observed Behavior
T1566.002	Phishing: Spearphishing Link	Personalized lure emails with links to attacker-controlled or compromised redirect infrastructure.
T1528	Steal Application Access Token	Abuse of OAuth 2.0 Device Authorization Grant to obtain a valid Microsoft 365 access/refresh token without credential theft.
T1204.001	User Execution: Malicious Link	Victim interaction with the phishing link/attachment triggers the device-code redirect chain.
T1114 / T1114.003	Email Collection	Post-compromise search and exfiltration of email content via Microsoft Graph API.
T1098.002 / T1136	Account Manipulation / Create Account (Inbox Rules)	Creation of malicious inbox rules for persistence and silent mail exfiltration.

Technique ID	Technique Name	Observed Behavior
T1550	Use Alternate Authentication Material (Token Replay)	Reuse of stolen access/refresh tokens from attacker infrastructure post-compromise.
T1098	Account Manipulation (Device Registration / PRT)	Registration of a new device to obtain a Primary Refresh Token for long-term persistence.
T1087.004	Account Discovery: Cloud Account	Microsoft Graph reconnaissance to map organizational structure and identify high-value accounts.
T1584 / T1583	Compromise/Acquire Infrastructure	Use of Railway.com, Cloudflare, DigitalOcean, Vercel, and AWS Lambda for disposable, trusted-platform infrastructure.

Indicators of Compromise (IoC)

Indicator	Type	Notes
`162.220.232.0/24`	IP range (Railway.com)	Threat-actor infrastructure observed issuing device-code sign-ins.
`162.220.234.0/24`	IP range (Railway.com)	Threat-actor infrastructure observed issuing device-code sign-ins.
`89.150.45.0/24`	IP range (HZ Hosting)	Threat-actor infrastructure observed issuing device-code sign-ins.
`185.81.113.0/24`	IP range (HZ Hosting)	Threat-actor infrastructure observed issuing device-code sign-ins.
`/api/device/start/ , /start*/`	Backend endpoint path	Attacker proxy endpoint that requests a live device code from Microsoft on victim click.
`/state`	Backend endpoint path	Polling endpoint used by checkStatus() to detect completed authentication.
`X-Antibot-Token`	HTTP header	64-character hex value sent with the device-code start request; empty body, content-length: 0.
`graph-microsoft[.]com, portal-azure[.]com, office365-login[.]com`	Impersonating domain pattern	Examples of brand-impersonating / domain-shadowing lure domains.

Indicator	Type	Notes
a7b2-c9d4.office-verify[.]net (pattern)	Randomized subdomain pattern	Randomized alphanumeric subdomain prefix on impersonating parent domains.
*.vercel.app , *.workers.dev , AWS Lambda endpoints	Redirect infrastructure	Legitimate serverless platforms abused for redirect chains ahead of the final landing page.
navigator.clipboard.writeText	Client-side JS API call	Used to silently push the generated device code onto the victim's clipboard.
GetCredentialType	Microsoft endpoint (abused for recon)	Queried 10-15 days pre-attack to validate target email addresses are active in-tenant.
microsoft.com/devicelogin	Legitimate Microsoft URL (abused)	The genuine Microsoft page victims are directed to; presence alone is not malicious -- context and referrer matter.

Detection and Threat Hunting

The following Microsoft Sentinel / Advanced Hunting KQL queries (adapted from Microsoft's April 2026 analysis) can be used by SOC teams with Microsoft Defender / Entra ID logging enabled. Sample rules are provided to assist technical teams in threat hunting; customize field names and thresholds for your own SIEM/EDR environment.

Detect interrupted-then-successful sign-in pattern (victim pausing to enter code)

```
EntraIdSigninEvents | where ErrorCode in (0, 50199) | summarize ErrorCodes = make_set(ErrorCode) by AccountUpn, CorrelationId, SessionId, bin(Timestamp, 1h) | where ErrorCodes has_all (0, 50199)
```

Detect device-code authentication from known-malicious IP ranges

```
EntraIdSigninEvents | where Call has "Cmsi:cmsi" | where IPAddress has_any ("162.220.232.*", "162.220.234.*", "89.150.45.*", "185.81.113.*")
```

Correlate suspicious URL clicks with risky sign-ins shortly after

```
let suspiciousUserClicks = materialize(   UrlClickEvents    | extend AccountUpn = tolower(AccountUpn)    | project ClickTime = Timestamp, ActionType, UrlChain, NetworkMessageId, Url, AccountUpn); let interestedUsersUpn = suspiciousUserClicks | where isnotempty(AccountUpn) | distinct AccountUpn; EntraIdSigninEvents | where ErrorCode == 0 | where AccountUpn in~ (interestedUsersUpn) | where RiskLevelDuringSignin in (10, 50, 100) | extend AccountUpn = tolower(AccountUpn) | join kind=inner suspiciousUserClicks on AccountUpn | where (Timestamp - ClickTime) between (-2min .. 7min) | project Timestamp, ReportId, ClickTime, AccountUpn, RiskLevelDuringSignin, SessionId, IPAddress, Url
```

Detect suspicious post-compromise device registration (PRT persistence)

```
CloudAppEvents | where AccountDisplayName == "Device Registration
Service" | extend ApplicationId_ =
tostring(ActivityObjects[0].ApplicationId) | extend DeviceName =
tostring(parse_json(tostring(RawEventData.ModifiedProperties))[1].NewValu
e) | extend UserPrincipalName = tostring(RawEventData.ObjectId) | project
TimeGenerated, ApplicationId_, DeviceName, UserPrincipalName
```

Detect malicious inbox-rule creation following compromise

```
CloudAppEvents | where ApplicationId == "20893" // Microsoft Exchange
Online | where ActionType in ("New-InboxRule","Set-InboxRule","Set-
Mailbox","Enable-InboxRule","UpdateInboxRules") | where
isnotempty(IPAddress) | mv-expand ActivityObjects | extend name =
parse_json(ActivityObjects).Name, value =
parse_json(ActivityObjects).Value | where name == "Name" | extend
RuleName = value
```

Network / proxy detection guidance

- Traditional URL/domain-reputation scanning will not detect the AES-GCM "ghost" payload -- it is encrypted at the time network tools inspect it. SOCs should prioritize browser-telemetry or endpoint-based inspection (e.g., browser extensions/EDR that observe the rendered DOM, clipboard writes, and post-render Fetch/XHR calls) over URL/domain blocklists alone.
- Alert on any outbound request to a path matching /api/device/start* or /state originating from a webpage that is not a known, approved application.
- Alert on unexpected clipboard-write events immediately followed by navigation to microsoft.com/devicelogin.
- Flag device code authentications for user accounts or applications that have never previously used this flow, and treat first-time device code usage as a high-priority triage item.
- Review Entra ID sign-in logs for the Authentication Flow = Device Code Flow field and cross-reference against a documented allow-list of approved device-code use cases (conference-room hardware, CI/CD, managed IoT)

Mitigation and Recommended Actions

Priority action: Block Device Code Authentication Flow via Microsoft Entra ID Conditional Access for all users/apps, except documented exceptions (meeting-room hardware, CI/CD, managed IoT) — this single control stops the threat entirely. If it can't be fully blocked, restrict polling to corporate IP ranges.

Harden authentication: Require phishing-resistant MFA (FIDO2, Authenticator passkey) — avoid SMS/voice MFA. Block legacy auth protocols. Enable Defender for Office 365 anti-phishing/Safe Links. Enforce sign-in risk policies for auto re-auth/block.

Detect and monitor: Run the Section 9 KQL queries against sign-in/audit logs and URL-click telemetry. Deploy browser/endpoint telemetry to catch the "ghost" payload (DOM, clipboard, network calls) — network gateways can't see it. Watch for anomalous Graph API activity and new device/PRT registrations within 10–15 minutes of sign-in.

Train users: Any "enter this code" prompt from an unsolicited email is a red flag — even on a genuine Microsoft domain. Users should cancel and report device-code prompts they didn't personally start, and report suspicious emails to IT/security instead of clicking.

If compromise is suspected: Revoke refresh tokens and force re-auth immediately; also disable the account, since old access tokens can stay valid up to 1 hour after revocation. Remove malicious inbox rules. Revoke any newly registered devices/PRTs. Check Graph API logs for data exfiltration or recon. Assess business-email-compromise/wire-fraud risk and notify affected counterparties.

References

- The Hacker News, "New Ghost Phishing Wave Is Breaking Traditional Email Security," 8 July 2026. https://thehackernews.com/2026/07/new-ghost-phishing-wave-is-breaking.html
- Microsoft Defender Security Research Team, "Inside an AI-enabled device code phishing campaign," Microsoft Security Blog, 6 April 2026. https://www.microsoft.com/en-us/security/blog/2026/04/06/ai-enabled-device-code-phishing-campaign-april-2026/
- Huntress, "Riding the Rails: Threat Actors Abuse Railway.com PaaS as Microsoft 365 Token Attack Infrastructure," March 2026. https://www.huntress.com/
- Sekoia Threat Detection & Research, "New widespread EvilTokens kit: device code phishing as-a-service -- Part 1," 30 March 2026. https://blog.sekoia.io/
- Cloud Security Alliance, analysis of EvilTokens PhaaS and OAuth 2.0 Device Authorization Grant abuse, 2026. https://cloudsecurityalliance.org/
- Push Security, device-code phishing detection research and 2026 SaaS attack telemetry. https://pushsecurity.com/blog/detecting-and-blocking-phishing-attacks-in-the-browser
- Microsoft Threat Intelligence, "Storm-2372 conducts device code phishing campaign," February 2025; Proofpoint reporting on TA4903, March 2026. https://www.microsoft.com/en-us/security/blog/
- MITRE ATT&CK, Technique T1528, "Steal Application Access Token." https://attack.mitre.org/techniques/T1528/
- BGD e-GOV CIRT, "Emerging Phishing Attack on Cyber Space of Bangladesh," 12 January 2025. https://www.cirt.gov.bd/phishing-awareness-bd-jan25/
- Microsoft Learn, "OAuth 2.0 device authorization grant flow." https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-device-code