DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

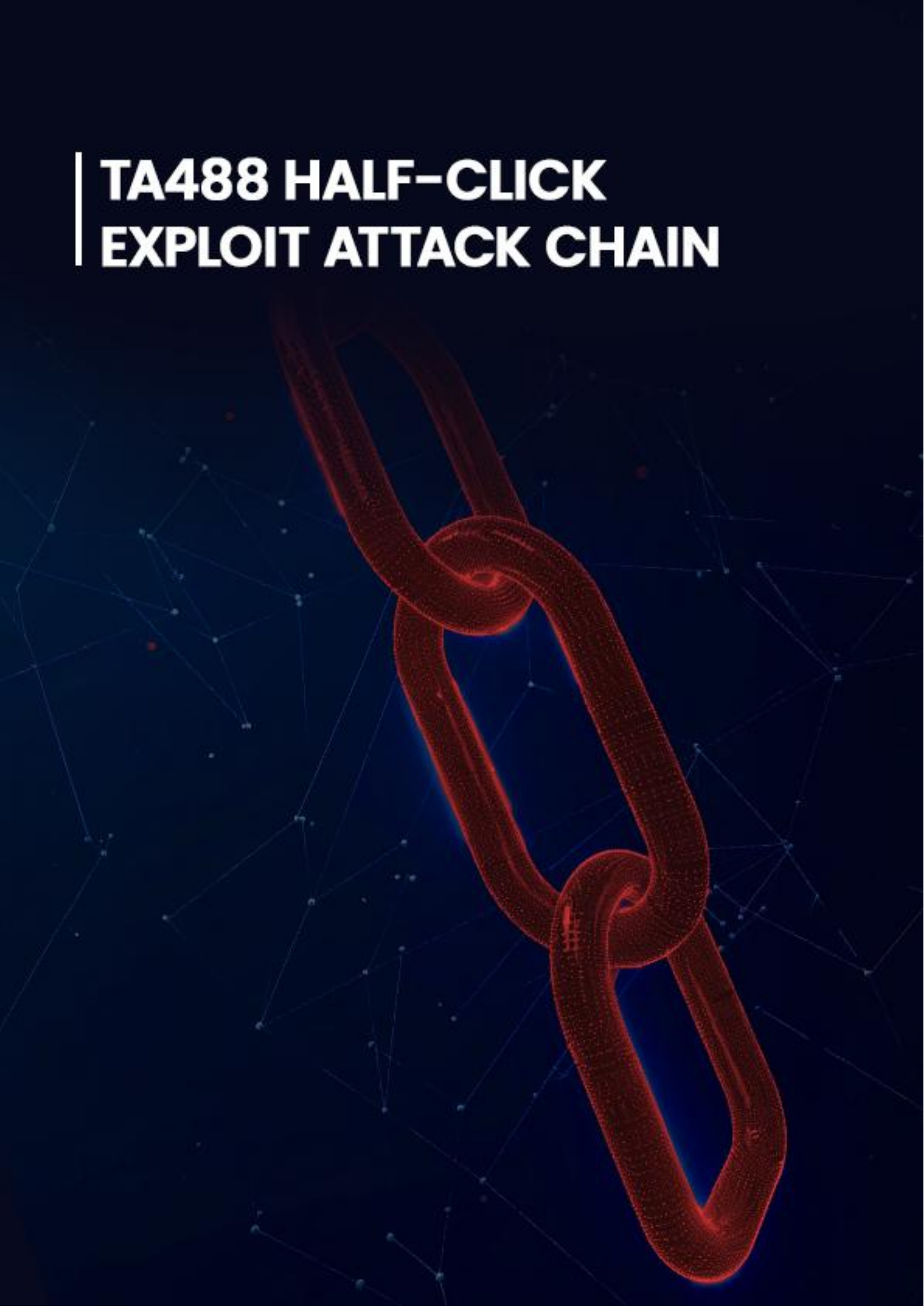
DSCI THREAT INTELLIGENCE AND RESEARCH INITIATIVE

THREAT REPORT

AUGUST 2026

Contents

TA488 HALF-CLICK EXPLOIT ATTACK CHAIN



TA488 Half-Click Exploit Attack Chain

Overview

On 22 July 2026, TA488 launched a new exploitation wave abusing CVE-2026-42897, a cross-site scripting (XSS) flaw in Outlook Web Access (OWA) caused by improper server-side sanitization of HTML in message bodies. The campaign is a "half-click" exploit due to which simply opening/viewing the email in OWA is sufficient to trigger code execution and action like clicking on a link or opening an attachment is required.

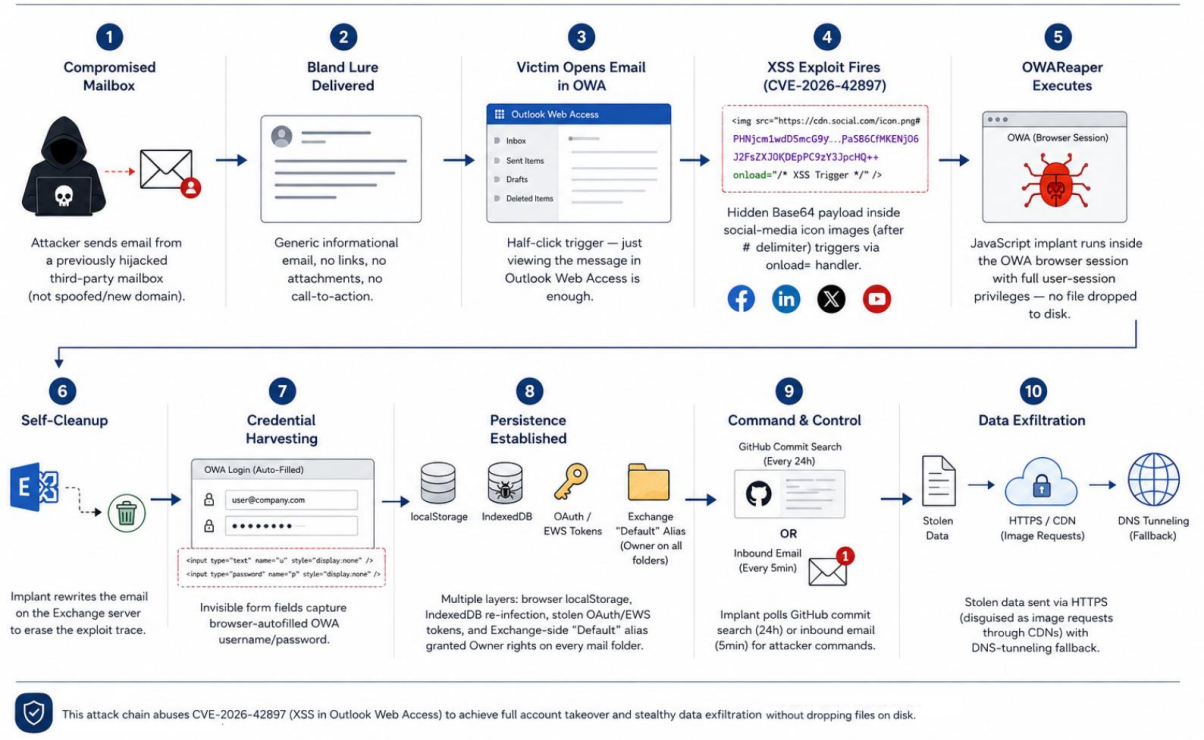
Successful exploitation deploys OWAReaper, a previously undocumented, purely browser-resident JavaScript implant that operates entirely within the OWA reading-pane context. It leaves no host-based artifacts, abuses legitimate OWA client-side storage and Exchange Web Services (EWS) functionality for persistence, and supports dual C2 channels (GitHub commit search and inbound email) and dual exfiltration channels (HTTPS via image-CDN proxying, and DNS label tunnelling as fallback).

Analysts assess with moderate confidence that TA488 may have used CVE-2026-42897 as a zero-day: attacker infrastructure was registered in March 2026, roughly two months before Microsoft's out-of-band patch.

Vulnerability details

Field	Detail
CVE ID	CVE-2026-42897
Affected Products	Microsoft Exchange Server - Outlook Web Access (OWA)
Vulnerability Type	Cross-Site Scripting (XSS) - CWE-79 (Improper Neutralization of Input)
Root Cause	The Exchange/OWA server does not adequately sanitize HTML content in inbound message bodies before rendering them in the reading pane
Trigger Condition	Victim opens or previews the email in OWA - no click, link, or attachment interaction required ("half-click" exploit)
Impact	Arbitrary JavaScript execution in the victim's authenticated OWA browser session

Attack Chain



Step 1 – Initial Access via Compromised Mailboxes

TA488 sends emails from previously compromised third-party mailboxes rather than freshly registered or spoofed infrastructure, improving deliverability and reducing reputation-based filtering hits.

Step 2 – Lure Delivery

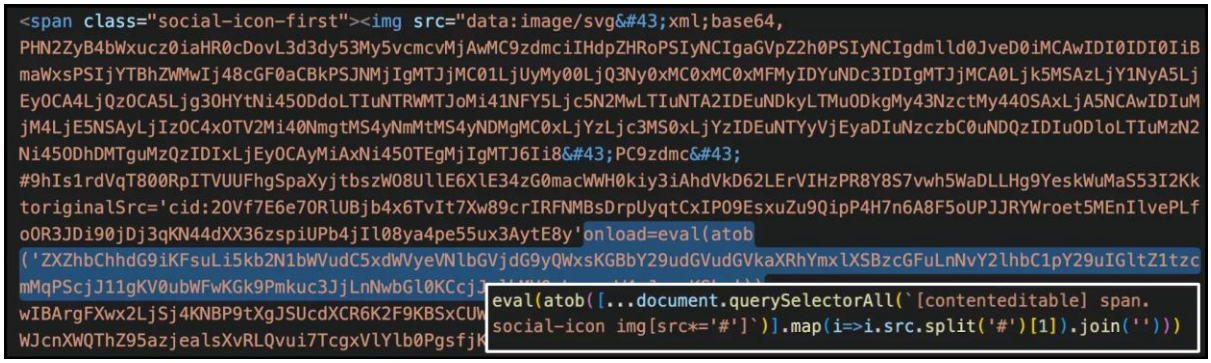
The email body is deliberately kept bland and low signal. It is framed as an informational content on subjects like gas markets, tourism indicators, semiconductor supply chains, public health surveillance, or nuclear energy fleet status. There is no call-to-action, no visible link, and no attachment, which is intended so that recipient quickly scans and discards the message as unimportant rather than flag or report it. Message volume and sector breadth in this wave were unusually high for TA488, likely to blend into bulk/spam traffic.

Step 3 – Victims Open the Email in OWA

The action of victims opening or reading the email in Outlook Web Access is sufficient to initiate exploitation because the exploit activates on message render/preview; no additional click is required (the ***"half-click"*** characteristic).

Step 4 – Exploit Triggers (CVE-2026-42897)

OWA's insufficient HTML sanitization allows the embedded ***"onload="*** handler in the lure's social media icon images to execute. The exploit payload is Base64-encoded and hidden after a **# delimiter** in the image reference. Browsers stop parsing the image at this character, making the trailing data invisible to normal rendering and filtering, while still allowing the malicious loader to retrieve and execute it. (see Section 3 for the underlying vulnerability mechanics).



Source: Proofpoint

Step 5 – Payload Reassembly and Execution

The ***"onload="*** handler (loader) parses the message DOM, extracts the Base64 fragments, reassembles and decodes them, and executes the result as JavaScript directly inside the OWA reading-pane browser context. This results in arbitrary script execution with the same privileges as the logged-in OWA user resulting in a session hijack inside the browser, without needing credentials, a malicious link, or an attachment.

This delivered script is the **OWAReaper implant**. It runs entirely client-side in the browser. No binary is dropped to disk, and no host-based EDR-visible artifact is created.

Step 6 – Initial Implant Actions/Self-Cleanup

Upon execution, OWAReaper:

- Uses Outlook APIs to rewrite the delivery email on the exchange server, stripping out the exploit content to reduce forensic evidence and avoid re-triggering or detection on re-open.
- Disables OWA pop-up blocking interference and right-click context menus while it runs.
- Generates a unique per-target session key.
- Collects the victim's email address, username, and OWA configuration/settings.

Step 7 - Credential Harvesting

OWAReaper creates two invisible "<*input*>" DOM elements positioned off-screen (-9999px/-9998px) and waits for the browser's native autofill feature to populate them with the victim's saved OWA username and password, then silently reads the values. This exact technique and offset is shared with TA488's earlier **ZimReaper** payload used against Zimbra webmail, reinforcing attribution to TA488.

Step 8 - Persistence Establishment (Multiple, Layered Mechanisms)

OWAReaper installs multiple, layered persistence mechanisms so that access survives credential rotation and even full device re-imaging.

1. **localStorage abuse:** OWAReaper writes an AES-encrypted copy of itself with a decryption wrapper into the browser's *localStorage*, nested under legitimate *PageDataPayload.OwaUserDefaultSettings* key (used by *OwaFrontendSyncState* sync-restore flow). As a result, every time the victim opens an OWA tab, native OWA sync logic re-executes OWAReaper automatically.

2. **Add-in/EWS token theft:** Enumerates installed outlook add-ins (valid in both OWA and the thick client) with *ReadWriteMailbox* permission. Abuse them to call *GetClientAccessToken* and steal OAuth/EWS access tokens while they are present.

3. **Server-side folder permission abuse:** It will call *UpdateFolder* to grant Owner-level permission on every mail folder to the tenant's low-privilege "Default" alias. This effectively exposes the full mailbox to any authenticated organization account, independent of the original victim endpoint. This persistence lives on server-side — credential rotation, or even full device replacement will not remove attacker access; remediation requires explicit permission-grant removal on the exchange server itself.

4. **IndexedDB/offline-cache re-infection:** Adds a hidden <iframe> to cached messages stored in OWA's offline IndexedDB message cache and enables caching. The iframe is re-executed each time the victim opens the poisoned message from the offline cache, meaning the implant can re-infect a freshly re-imaged endpoint as soon as the user's OWA offline cache re-syncs the old message.

Step 9 - Command and Control

Once persistence is in place, OWAReaper begins polling for operator instructions via one of two channels.

Channel	Mechanism	Polling Interval
GitHub Commit Search	Queries GitHub's Commit Search API for commit messages containing the target's email address; payload is AES-CTR decrypted with a hardcoded key + a per-session AES key	Every 24 hours
Inbound email	Parses IndexedDB-cached message bodies for a {target_email}{space}{Base64text} pattern	Polls offline cache every 5 minutes (looks back 30 minutes)

Command types

- code – replace OWAReaper's entire toolkit
- domn – rotate C2 domains
- cmnd – executes arbitrary one-time code via eval()

OWAReaper cannot pre-emptively hide or flag command-bearing emails, so operators likely disguise command emails using the same bland lure style as the initial delivery message.

Step 10 - Data Exfiltration (Dual-Protocol)

Primary — HTTPS: Outbound data such as telemetry, stolen credentials, session/token data, command output, etc. is assembled as structured key-value JSON by a send() dispatcher, encrypted via AES-CTR in _buildPackets(), and sent through _httpCascade, disguised as image asset requests *(/assets/v1_<base64_aes_data>)* proxied through legitimate third-party image CDNs to blend into normal traffic:

- images.weserv.nl (Weserv image cache/resize)
- i3.wp.com (WordPress image CDN)
- slack-imgs.com (Slack image CDN)

These CDNs ultimately relay to TA488 infrastructure at acocdn[.]com. If all CDN proxies fail, OWAReaper falls back to a direct HTTP GET to the C2. Notably, file contents are sent unencrypted as the raw POST body (only the URI metadata is AES-CTR encrypted) for files such as: ews_extensions_debug.json (add-in metadata dump), msanalytics.json (raw OWA session blob from /owa/sessiondata.ashx), cmnd_[HASH]_[FILENAME] (remote command output), and poison_wizard_error_dom.html (DOM snapshot on persistence failure).

Source: Proofpoint

Fallback — DNS Label Tunneling: If HTTPS delivery fails, data is AES-CTR encrypted and then Base32-encoded using a restricted alphanumeric alphabet (0–9, a–v) which is then split randomly into 2–4 substrings and embedded as subdomain labels in DNS A-record queries to an attacker-controlled domain.

Source: Proofpoint

Predecessor (ZimReaper): Unlike ZimReaper, OWAReaper does not perform bulk email exfiltration — assessed as a deliberate OPSEC improvement to reduce detectable data volume and network noise.

Detection & Remediations

- **Patch:** It is confirmed that the 9 June 2026 exchange security update addressing CVE-2026-42897 is installed (not just eligible/available) – patch adoption lag is most likely the reason that this campaign still succeeded in July. Also, it is confirm that 14 May 2026 EEMS mitigation remains active as per Microsoft's guidance. See the exchange team blog: Addressing Exchange Server May 2026 Vulnerability (CVE-2026-42897).
- **Audit and revoke EWS/OAuth tokens:** Revoke add-ins with ReadWriteMailbox permission that may have been abused via GetClientAccessToken.
- **Audit exchange folder permissions:** Audit the permissions across all mailboxes; remove any unexpected **Owner-level grants to the "Default" alias** — this is the primary server-side persistence mechanism and will survive endpoint remediation.
- **Clear browser-side persistence:** Check for browser-side persistence on affected/exposed endpoints:
  - Clear OWA's offline IndexedDB cache (owa_offline_db).
  - Clear the *PageDataPayload.OwaUserDefaultSettings* key in browser localStorage.
- **Network detection/blocking:** Alert on or block outbound connections to known C2 domains.
- **Credential hygiene:** Rotate credentials for any account that accessed OWA during the exposure window. Although this alone is not sufficient to evict the actor due to server-side persistence.
- **Email filtering:** Flag/inspect inbound HTML messages containing embedded social-media-style icon images with Base64 data following # delimiters, and messages matching the {email}{space}{base64} command pattern.

Indicators of Compromise (IOCs)*

Indicator	Type	Description
asecdns[.]com	Domain	OWAReaper C2
acocdn[.]com	Domain	OWAReaper C2
dnsrecursive[.]eu	Domain	OWAReaper C2
tdndns[.]com	Domain	OWAReaper C2
6897b649f29e54d8910459963bbf94ed5c7a4fe66a56bc5962540b226b8e48c4	SHA256	HTML message body containing exploit and OWAReaper payload

**The Indicators of Compromise (IoCs) listed above have been compiled from publicly available threat intelligence published by Proofpoint and are included for reference purposes.*

Legitimate services abused:

- images.weserv.nl
- i3.wp.com
- slack-imgs.com

CVE-2025-69414

CVE-2025-69414

Overview

On 12 August 2026, security researcher Nightmare Eclipse dropped a fully working exploit for Microsoft Defender itself which is a flaw in Defender's own scanning engine that lets an attacker use it as a launchpad to seize full control of the machine.

This exploit is called ShieldBreak and tracked as CVE-2026-69414, which allows an attacker who already has limited access (normal user access) of a windows machine to escalate itself to SYSTEM-level privileges which is the highest privilege of windows.

Timeline

Date	Event
16 June 2026	Microsoft acknowledges RoguePlanet (CVE-2026-50656), a race-condition elevation of privilege (EoP) flaw in the Malware Protection Engine
8 July 2026	Microsoft ships a patch for RoguePlanet
12 August 2026	Nightmare Eclipse publishes ShieldBreak, a working PoC demonstrating a full bypass of the RoguePlanet patch, assigned CVE-2026-69414, released after August Patch Tuesday
14 August 2026	Microsoft publishes the CVE-2026-69414 advisory, confirming it as an EoP in the Malware Protection Engine and stating a security update is in progress
15 August 2026	CISA adds CVE-2026-69414 to its KEV catalog

About Vulnerability

The flaw lives in the Microsoft Malware Protection Engine which is the actual scanning brain behind Microsoft Defender. This engine runs with very high privileges as it needs to inspect files across the entire system, including files that a normal user account shouldn't be able to touch directly. That's also the whole point of an AV engine, and this is exactly what makes it a valuable target as if you can trick a highly-privileged process into doing something on your behalf, you inherit its privileges.

ShieldBreak achieves this by targeting a specific corner of Windows which is **cloud-file hydration**. When you use a cloud-sync service like OneDrive, files that live in the cloud often appear on your local disk as lightweight "placeholders." The moment a process or anything else needs to read that file, Windows fetches ("hydrates") the real content through a system called the Cloud Filter API (CFAPI).

The trick here is that defender's scan engine gets involved in that hydration process too, since a file arriving from the cloud needs to be scanned. ShieldBreak exploits a race condition which is a timing gap during this process. In this process, the attacker can:

1. Register their own callback into the CFAPI hydration flow which is something a standard user account is permitted to do.
2. Time that callback to interfere with the actual bytes Defender receives while the file is being hydrated effectively swapping in different content after the engine has already decided which file it's about to scan, but before it has finished reading it.
3. Combine and manipulate it with lower-level Windows plumbing (filesystem and Object Manager internals) to steer the engine toward a wrong file object which is controlled by the attacker.

Now, the scanning engine ends up interacting with an attacker-supplied content while still believing that it's doing its normal privileged job. As this job runs with SYSTEM-level privileges, it does whatever the attacker just fed it. The researcher claims a **100% success rate** on Windows 11 25H2 and Windows Server 2025.

The attacker now holds full SYSTEM privileges on the host; complete administrative control, independent of the original low-privilege account's actual permissions. Now, an attacker can dump credentials (e.g., LSASS), disable further security tooling, establish persistence, and move laterally in the developer workstations (pivot to CI/CD, credential stores, internal tooling) and on Windows Server 2025 infrastructure (domain controllers, build agents, application servers).

This is a **local** attack, not a remote one. An attacker needs some starting foothold on the machine first, a standard user session is enough, no admin rights required, and no user interaction is needed beyond that initial access. The public testing shows that exploit only works when Microsoft Defender's engine is actively running as the AV provider. If Defender is disabled, or a different antivirus product has taken over as the registered scanner, this particular chain doesn't fire.

Relation with Older Vulnerability

Back in June 2026, the same researcher had already found a very similar problem which was called **RoguePlanet (CVE-2026-50656)**, which is also a race-condition privilege escalation in the same scanning engine. Microsoft patched RoguePlanet on 8 July 2026 but it didn’t solve the problem completely as the fix closed one specific timing window that led to the success of the race condition. It did not remove the deeper architectural issue that made the engine exploitable in the first place. ShieldBreak reaches the same end result - SYSTEM-level compromise - through a different technical path (the CFAPI hydration/Object Manager combination). In security terms, this can be called patch bypass.

Exploitation in Real World

Currently, there is no confirmed evidence of ShieldBreak being used in real-world attacks; however, this is a public security-research disclosure, which belongs to high-risk category. Treat the risk as practical and imminent rather than actively exploited in the wild, as reliable

PoCs for privilege-escalation vulnerabilities have historically been weaponized within days to weeks of public release, rather than months.

What's Affected

Platform	Exposure
Windows 11 (all supported versions, including 25H2 and Canary)	Affected; confirmed working exploitation on 25H2/Canary
Windows 10 (all supported versions, Defender enabled)	Affected
Windows Server 2025	Affected; confirmed working exploitation — treat as high priority
Windows Server 2022 / 2019	Affected wherever Defender is the active scanning engine
Machines running a third-party AV as the *active* scanner	Not directly affected by this specific chain

Detection and Mitigations

No patch has been released yet for this, so realistic approach is layering several compensating controls. This means that even if one fails, exploitation gets harder or gets caught:

- **Enable Defender cloud protection and block at first sight:** Cloud-based heuristics update faster than local signatures and increase the odds of catching suspicious CFAPI-related scan activity.
- **Trim local account privileges wherever possible:** ShieldBreak needs local access to start with so fewer accounts with local logon rights mean a smaller attack surface.
- **Turn on controlled folder access (part of exploit guard):** This limits what unauthorized processes can write to, which constrains what an attacker can do even after escalating.
- **Enable Attack Surface Reduction (ASR) rules:** These help block the follow-on moves attackers typically make after gaining SYSTEM privileges like credential theft from LSASS, script-based lateral movement, etc.
- **Watch for anomalous SYSTEM-level process activity**: Particularly check for cmd.exe, powershell.exe, or wscript.exe spawning as children of MsMpEng.exe (the Defender scan engine process) without a legitimate reason.
- **Isolate self-hosted CI/CD build agents:** Isolate them from general developer workstations to limit blast radius if one machine is compromised and escalated.
- **Watch Microsoft's official advisory:** Keep a close watch for CVE-2026-69414 patch release and apply the fix immediately the moment it's released as this is not a vulnerability where a delayed rollout is acceptable.

Detection Guidance: Since no specific signature is published yet, so detection currently should focus on behavior rather than blocking the technique itself:

- **Check for alerts** on SYSTEM-privileged processes spawned unexpectedly from MsMpEng.exe.
- **Correlate unusual CFAPI/cloud-hydration activity** with a privilege change on the same host within a short time window.
- **Flag any transition** from a standard user session to SYSTEM-level activity that doesn't line up with a legitimate elevation event (UAC prompt, scheduled task launch, service start). This is a useful general heuristic for local privilege escalation, not just this specific bug.

References

1. Proofpoint: Cleaning Out Inboxes: TA488 Comes for Outlook with Another Half-Click Exploit. https://www.proofpoint.com/us/blog/threat-insight/cleaning-out-inboxes-ta488-comes-outlook-another-half-click-exploit
2. Bleeping Computer: Russian hackers exploit Exchange OWA zero-day for long-term mailbox access: https://www.bleepingcomputer.com/news/security/russian-hackers-exploit-exchange-owa-zero-day-for-long-term-mailbox-access/
3. The Hacker News: Russian Hackers Exploit Microsoft OWA Flaw to Keep Mailbox Access After Credential Rotation: https://thehackernews.com/2026/07/russian-hackers-exploit-microsoft-owa.html
4. National Vulnerability Database: CVE-2026-42897: https://nvd.nist.gov/vuln/detail/cve-2026-42897
5. Microsoft Exchange Team Blog: Addressing Exchange Server May 2026 Vulnerability (CVE-2026-42897): https://techcommunity.microsoft.com/blog/exchange/addressing-exchange-server-may-2026-vulnerability-cve-2026-42897/4518498
6. Malwarebytes: ShieldBreak bypasses Microsoft's patch for earlier Defender flaw: https://www.malwarebytes.com/blog/bugs/2026/08/shieldbreak-bypasses-microsofts-patch-for-earlier-defender-flaw
7. Malwarebytes: Microsoft working on a fix for RoguePlanet, a flaw that grants full PC control: https://www.malwarebytes.com/blog/news/2026/06/microsoft-working-on-a-fix-for-rogueplanet-a-flaw-that-grants-full-pc-control
8. BleepingComputer: New Microsoft Defender 'ShieldBreak' zero-day grants SYSTEM privileges: https://www.bleepingcomputer.com/news/security/new-microsoft-defender-shieldbreak-zero-day-grants-system-privileges/
9. SecurityWeek: Nightmare Eclipse Drops Windows Zero-Day Exploit 'ShieldBreak': https://www.securityweek.com/nightmare-eclipse-drops-windows-zero-day-exploit-shieldbreak/

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

ABOUT DSCI

Data Security Council of India (DSCI), set up by Nasscom, is a not-for-profit industry body and think tank on data protection, cyber security, and critical technologies in India. DSCI is committed to making cyberspace safe, secure, and privacy-assured, and to advancing critical technologies vital for national security through policy advocacy, thought leadership, capacity building, and outreach initiatives. DSCI engages with the government, regulators, defence, law enforcement agencies, embassies, industry members, and academia to position India as a leader in the global digital ecosystem.

Data Security Council of India
4th Floor, Nasscom Campus, Plot No. 7-10, Sector 126, Noida, UP-201303

data-security-council-of-india/ DSCI_Connect dsci.connect
dsci.connect dscivideo security chips