Analyse forensique avancée des vecteurs d'exfiltration furtive via le sous-système Desktop Window Manager (dwm.exe)

L'évolution de l'architecture graphique de Windows, depuis l'introduction du Windows Display Driver Model (WDDM), a transformé le processus Desktop Window Manager (DWM) d'un simple gestionnaire de fenêtres en un moteur de composition complexe, riche en fonctionnalités et opérant avec des privilèges de haut niveau. Dans le cadre d'une analyse de menaces persistantes avancées (APT), il apparaît que le processus dwm.exe ne constitue plus seulement un composant de l'interface utilisateur, mais une surface d'attaque critique pour l'exfiltration de données visuelles sans trace disque. Cette étude déconstruit les mécanismes par lesquels un attaquant peut détourner les API de composition pour transformer ce processus légitime en un enregistreur d'écran passif, exploitant la mémoire vive comme zone de transit primaire avant une exfiltration fragmentée vers des infrastructures de confiance.

Architecture systémique et surface d'attaque de DirectComposition

Le Desktop Window Manager est le pivot central de l'affichage sous Windows 10 et 11. Contrairement aux versions antérieures de Windows où les applications dessinaient directement sur l'écran via GDI (Graphics Device Interface), les versions modernes utilisent DirectComposition. Cette API, introduite avec Windows 8, permet une composition de bitmaps haute performance avec des transformations, des effets et des animations.[1] Le fonctionnement de DirectComposition repose sur une structure client-serveur complexe : une application (le client) utilise dcomp.dll pour envoyer des commandes à win32kbase.sys via des appels système exportés par win32u.dll.[2] Les données sont ensuite transmises au processus dwm.exe (le serveur) via une section de mémoire partagée pour finaliser le rendu graphique.[2]

Cette architecture de mémoire partagée crée une surface d'attaque massive. Puisque les utilisateurs avec des privilèges restreints doivent pouvoir interagir avec le processus DWM pour rendre leurs fenêtres, il existe un canal de communication inter-processus (IPC) qui traverse les frontières de sécurité.[3] Le processus DWM, s'exécutant sous l'identité DWM-1 ou SYSTEM avec un niveau d'intégrité élevé, devient une cible de choix pour l'escalade de privilèges et la persistance.[3]

Mécanismes des appels système DirectComposition

La manipulation des buffers de rendu nécessite une compréhension chirurgicale des appels système (syscalls) NtDComposition. Ces fonctions sont le point d'entrée pour toute injection de commandes de composition dans le flux de rendu du système.

Appel Système	Description Technique	Rôle dans l'Exploitation
NtDCompositionCreateChannel	Initialise un canal de communication et mappe une adresse mémoire partagée.	Permet d'obtenir le pMappedAddress pour injecter des commandes malveillantes.[1]
NtDCompositionProcessChannelBatchBuffer	Envoie par lots plusieurs commandes DirectComposition pour traitement par le noyau.	Utilisé pour sérialiser des commandes manipulant les ressources de rendu.[1]
NtDCompositionCommitChannel	Déclenche l'exécution des commandes mises en lot et notifie le processus DWM.	Agit comme le déclencheur final pour le traitement du buffer potentiellement corrompu.[2]

L'exploitation de ces canaux permet non seulement de manipuler l'affichage, mais aussi d'accéder aux textures de toutes les fenêtres actives, car le DWM possède la vue d'ensemble du bureau complet. En injectant du code dans dwm.exe via des techniques de type Reflective DLL Injection, un attaquant peut intercepter les pointeurs vers ces textures avant qu'elles ne soient libérées de la mémoire vidéo ou de la RAM système.[1]

Analyse de la vulnérabilité CCommandBuffer et exploitation du tas

La sécurité du processus DWM repose sur la robustesse de sa bibliothèque centrale, dwmcore.dll. Des recherches récentes ont mis en lumière des vulnérabilités critiques de type débordement de tas (heap overflow) dans ce module, notamment la CVE-2024-30051.[4] Cette faille est symptomatique des erreurs de gestion mémoire dans les moteurs de rendu complexes. Elle réside dans la méthode CCommandBuffer::Initialize, où une discordance existe entre le calcul de la taille d'allocation et la taille de la copie de données.[4]

Le processus de vulnérabilité se manifeste lorsque le code appelle CD2DSharedBuffer::GetBufferSize pour déterminer la taille du buffer de commande entrant. L'allocation mémoire est effectuée selon une formule d'arrondi par division entière :

$$TailleAllocation = \left(\frac{TailleBuffer}{0x90} \right) \times 0x90$$

Cependant, l'opération memcpy subséquente utilise la TailleBuffer brute, sans l'opération d'arrondi.[4] Par exemple, si un attaquant fournit un buffer de $0x91$ octets, l'allocation ne sera que de $0x90$ octets, provoquant un débordement d'un octet contrôlé sur le tas.[4] Ce type de "heap overflow" millimétré permet de corrompre des objets adjacents dans la mémoire de dwm.exe.

Stratégies d'élévation de privilèges via DWM

L'exploitation réussie de telles vulnérabilités permet à un utilisateur sans privilèges d'exécuter du code avec l'identité du processus DWM, accédant ainsi à un niveau d'intégrité système.[4] Les acteurs de menaces comme QakBot ont été observés utilisant ces vulnérabilités de type "zero-day" pour l'escalade de privilèges.[7] Le processus d'exploitation suit généralement un schéma rigoureux :

1. **Interception d'API (Hooking) :** L'exploit installe des hooks sur RtlAllocateHeap, RtlCreateHeap, NtDCompositionCreateChannel et NtDCompositionCommitChannel pour capturer les handles de tas et les adresses mappées.[4]
2. **Aspersion du tas (Heap Spraying) :** Une technique d'aspersion est utilisée pour stabiliser la mémoire de dwm.exe et s'assurer que les objets cibles se trouvent à des emplacements prévisibles après le buffer vulnérable.[4]
3. **Déclenchement du débordement :** En envoyant des paquets spécifiquement formatés via NtDCompositionProcessChannelBatchBuffer, l'attaquant force le déclenchement de la faille dans dwmcore.dll.[4]

Une fois le contrôle du flux d'exécution obtenu, l'attaquant n'a plus besoin d'écrire de fichiers sur le disque. Le code malveillant réside uniquement en mémoire vive, ce qui rend les solutions antivirus traditionnelles inefficaces car elles se concentrent principalement sur les artefacts persistants.[6]

Détournement des buffers de rendu pour la capture d'écran passive

Une fois le processus dwm.exe compromis, il peut être transformé en un outil de surveillance sophistiqué. Au lieu d'utiliser des API de capture d'écran standard qui pourraient être surveillées par des outils d'Endpoint Detection and Response (EDR), l'attaquant détourne les buffers internes de DirectComposition ou de la Desktop Duplication API.[9]

Dans un scénario d'exfiltration furtive, la RAM est utilisée comme un buffer de capture massif. Étant donné que dwm.exe a déjà accès à toutes les textures des fenêtres pour les composer,

le malware peut simplement copier ces textures dans une zone de mémoire allouée dynamiquement. Cette zone, invisible pour le système de fichiers, peut atteindre des tailles considérables, entre 10 et 15 Go, agissant comme un enregistreur vidéo volatil.[10]

Gestion de la mémoire et pagination furtive

Stocker 15 Go de captures d'écran brutes en RAM sans provoquer de plantage système ou d'alertes de performance nécessite une gestion fine de la mémoire virtuelle. Le malware peut utiliser des techniques de "paging" manuel pour déplacer ces données visuelles vers le fichier de pagination (pagefile.sys) tout en gardant les pointeurs actifs dans l'espace d'adressage de dwm.exe.

Technique de Gestion Mémoire	Description Forensique	Avantage pour l'Attaquant
VirtualAlloc avec SEC_COMMIT	Allocation de mémoire virtuelle sans fichier de support immédiat.	Évite la création de fichiers temporaires sur le disque.[6]
Reflective DLL Injection	Chargement d'une DLL directement depuis la mémoire.	Ne laisse aucune trace dans la liste des modules chargés de manière standard.[6]
Process Hollowing	Remplacement du code d'un processus légitime par du code malveillant.	Masque l'activité malveillante sous le nom d'un processus système de confiance.[12]

L'utilisation de la mémoire vive comme buffer permet de contourner les journaux d'audit du système de fichiers. Tant que le système n'est pas redémarré, les preuves de la capture restent hautement volatiles et difficiles à détecter pour un analyste qui ne pratiquerait pas une analyse régulière de la mémoire vive.

Tunneling et exfiltration via des infrastructures de confiance

Le défi majeur pour une exfiltration de 15 Go de données visuelles est la détection réseau. Un transfert direct vers un serveur de commande et contrôle (C2) inconnu déclencherait immédiatement des alertes de volume de données. Pour pallier cela, les attaquants utilisent le

tunneling via des hôtes de confiance comme 1e100.net (Google) ou des services AWS.[13]

L'exfiltration est fragmentée en petits paquets, intégrés dans des flux de télémétrie légitimes. En utilisant des intervalles réguliers (heartbeats) toutes les 4 à 5 secondes, le malware synchronise l'envoi de fragments de captures d'écran avec les pics d'activité réseau attendus des services Windows.[13]

Corrélation entre pics mémoire et trafic réseau

Une analyse fine révèle une corrélation entre l'utilisation de la mémoire par le service DWM et les pics d'activité réseau vers des domaines Google ou Amazon. Lorsque dwm.exe capture une nouvelle frame, la consommation de RAM augmente. Peu après, un flux réseau crypté vers 1e100.net est observé, correspondant à l'envoi du fragment.[13]

Le choix de 1e100.net est stratégique. Ce domaine est omniprésent dans les environnements Windows pour les services de mise à jour, de recherche et de télémétrie de Google Chrome ou d'autres applications intégrées. De nombreuses politiques de sécurité d'entreprise autorisent ce trafic par défaut, ce qui permet au malware de se fondre dans la "pollution" réseau standard.[13]

Persistance "Memory-only" et évasion des logs

La persistance sans log est l'un des aspects les plus critiques des malwares modernes de type "Memory-only". En évitant d'écrire sur le disque, ils neutralisent les scanners de signatures et les outils de surveillance de l'intégrité des fichiers. QakBot, par exemple, décode ses chaînes de caractères uniquement au moment de l'exécution et les détruit immédiatement après usage, ne laissant que des artefacts éphémères en RAM.[8]

L'utilisation du moteur de rendu graphique pour la surveillance utilisateur permet au malware de fonctionner de manière totalement passive. Il n'a pas besoin de créer des hooks clavier ou souris classiques, qui sont souvent surveillés. En observant simplement les changements dans les buffers de dwmcore.dll, il peut enregistrer l'activité de l'utilisateur, y compris la saisie de mots de passe via des claviers virtuels ou la consultation de documents sensibles.

Manipulation des structures internes de processus

Pour maintenir cette invisibilité, le malware manipule les structures de données internes de Windows :

- **Modification des en-têtes PE :** Certains malwares altèrent les en-têtes PE en mémoire pour empêcher les outils forensiques de reconnaître les sections de code injecté.[8]
- **Détournement de l'IAT (Import Address Table) :** Les adresses des API légitimes sont remplacées par des adresses pointant vers le code malveillant, permettant une interception transparente des appels système.[12]

- **Nettoyage des traces de chargement :** Le malware peut se retirer lui-même de la liste InLoadOrderModuleList du PEB (Process Environment Block) pour ne pas apparaître lors d'une énumération standard des DLL chargées.[6]

Ces techniques forcent l'expert forensique à utiliser des méthodes de scan de bas niveau, comme le "pool tag scanning", pour identifier des zones de mémoire qui ne sont liées à aucun module légitime connu.[15]

Méthodologies forensiques pour l'extraction de textures volatiles

La détection de screenshots persistants dans la RAM de dwm.exe nécessite une approche chirurgicale. Les outils classiques de capture d'écran de mémoire comme le plugin screenshot de Volatility sont souvent limités aux versions anciennes de Windows et ne gèrent pas efficacement les structures de DirectComposition sous Windows 10/11.[16]

Acquisition et reconstruction des buffers graphiques

L'investigation doit se concentrer sur le dump spécifique des textures stockées dans l'espace d'adressage de dwm.exe. Cela implique plusieurs étapes complexes :

1. **Extraction du dump mémoire :** Utilisation d'outils comme procdump ou des dumpers de RAM complets pour capturer l'état de dwm.exe.[12]
2. **Identification des structures DirectComposition :** Analyse des objets Visual et des surfaces de composition. Ces structures contiennent des pointeurs vers les ressources bitmaps réelles.[2]
3. **Analyse des Pool Tags :** Recherche de tags spécifiques liés aux drivers graphiques et à DirectComposition pour localiser les buffers de textures orphelins ou cachés.[15]

Outil Forensique	Fonctionnalité	Application DWM
Volatility (psscan)	Scan des structures de processus via pool tags.	Identifie les processus cachés ou désinfectés qui pourraient héberger le malware.[15]
Rekall (analyze_struct)	Analyse récursive des structures de données en mémoire.	Permet de naviguer dans les structures non documentées de dwmcore.dll.[19]

mdls	Liste les attributs de métadonnées de fichiers.	Utile pour corréler des fichiers de log avec des activités de rendu suspectes.[14]

Reconstruction manuelle des preuves visuelles

La preuve ultime réside dans la reconstruction des textures. Puisque les données sont souvent stockées sous forme de bitmaps bruts (souvent en format 32-bit ARGB), un analyste peut extraire ces blocs de mémoire et les visualiser à l'aide d'outils de rendu brut. La présence de fragments reconnaissables de l'interface utilisateur, de documents sensibles ou de fenêtres de navigateur dans des zones de mémoire allouées par un module suspect au sein de dwm.exe constitue une preuve irréfutable d'exfiltration visuelle.[14]

Synthèse des vecteurs de menace et recommandations

L'utilisation de dwm.exe comme vecteur d'exfiltration représente l'apogée des techniques "Living off the Land". En exploitant un processus critique pour le fonctionnement du système, les attaquants s'assurent une persistance robuste et une discrétion maximale. Les vulnérabilités de type CVE-2024-30051 démontrent que même les composants les plus fondamentaux de Windows sont vulnérables à des attaques de précision.[4]

Pour contrer ces menaces, les organisations doivent adopter une posture de défense en profondeur qui ne repose pas uniquement sur la surveillance du disque. Une surveillance accrue des anomalies de mémoire dans les processus système critiques, combinée à une analyse comportementale du trafic réseau vers les infrastructures de confiance, est essentielle. La détection des heartbeats et la corrélation avec l'utilisation de la RAM par le DWM sont des indicateurs de compromission (IoC) de haute fidélité qui devraient être intégrés dans les stratégies de monitoring moderne.

La complexité croissante des API graphiques de Windows continuera d'offrir de nouvelles opportunités aux attaquants. Seule une compréhension profonde de l'ingénierie inverse des processus système et une capacité d'analyse forensique de la mémoire vive permettront de lever le voile sur ces opérations d'exfiltration furtives. Les experts en sécurité doivent rester sceptiques vis-à-vis des comportements "normaux" des processus système et développer des outils capables d'auditer en temps réel l'intégrité des structures de données en RAM, là où se cachent désormais les menaces les plus avancées.

Bryan Ouellette du Lichen-Collectives.

Ouvrages cités

1. Zero-day vulnerability in Desktop Window Manager (CVE-2021-28310) used in the wild, dernier accès : février 12, 2026, https://securelist.com/zero-day-vulnerability-in-desktop-window-manager-cve-2021-28310-used-in-the-wild/101898/
2. The Lost World of DirectComposition: Hunting Windows Desktop Window Manager Bugs, dernier accès : février 12, 2026, https://conference.hitb.org/hitbsecconf2023ams/materials/D1T1%20-%20The%20Lost%20World%20of%20DirectComposition-%20Hunting%20Windows%20Desktop%20Window%20Manager%20Bugs%20-%20Zhang%20WangJunjie.%20YiSheng%20He%20&%20Wenyue%20Li.pdf
3. The Lost World of DirectComposition: Hunting Windows Desktop Window Manager Bugs, dernier accès : février 12, 2026, https://conference.hitb.org/hitbsecconf2023ams/session/hunting-windows-desktop-window-manager-bugs/
4. Windows DWM Core Library Elevation of Privilege Vulnerability ..., dernier accès : février 12, 2026, https://www.coresecurity.com/core-labs/articles/windows-dwm-core-library-elevation-privilege-vulnerability-cve-2024-30051
5. Research on the CVE-2024-30051 privilege escalation vulnerability in the wild - Qianxin Threat Intelligence Center, dernier accès : février 12, 2026, https://ti.qianxin.com/blog/articles/public-secret-research-on-the-cve-2024-30051-privilege-escalation-vulnerability-in-the-wild-en/
6. Detection of Anomalous In-Memory Process based on DLL Sequence - Semantic Scholar, dernier accès : février 12, 2026, https://pdfs.semanticscholar.org/08a0/422931959a1a67c3b7b7320147a6232d9be3.pdf
7. Malware report for Q2 2024 — a quarterly review | Securelist, dernier accès : février 12, 2026, https://securelist.com/it-threat-evolution-q2-2024/113669/
8. QakBot reducing its on disk artifacts - Hornetsecurity – Next-Gen Microsoft 365 Security, dernier accès : février 12, 2026, https://www.hornetsecurity.com/en/blog/qakbot-reducing-its-on-disk-artifacts/
9. The Threat of Screenshot-Taking Malware: Analysis, Detection and ..., dernier accès : février 12, 2026, https://www.scribd.com/document/590215234/The-Threat-of-Screenshot-Taking-Malware-Analysis-Detection-and-Prevention
10. Report of
11. cs0-002 2 | PDF | Information Security - Scribd, dernier accès : février 12, 2026, https://www.scribd.com/document/814486184/cs0-002-2
12. (Security and Cryptology 10453) Marc Dacier, Michael Bailey, Michalis Polychronakis, Manos Antonakakis (Eds.)-Research in Attacks, Intrusions, And Defenses_ 20th International Symposium, RAID 2017, At (1) - Scribd, dernier accès : février 12, 2026, https://www.scribd.com/document/373070092/Security-and-Cryptology-10453-

Marc-Dacier-Michael-Bailey-Michalis-Polychronakis-Manos-Antonakakis-Eds-Research-in-Attacks-Intrusions-And-Def
13. UNIVERSIDADE FEDERAL DO PARANÁ MARCUS FELIPE BOTACIN ON THE MALWARE DETECTION PROBLEM - Acervo Digital UFPR, dernier accès : février 12, 2026, https://acervodigital.ufpr.br/xmlui/bitstream/handle/1884/85292/R%20-%20T%20-%20MARCUS%20FELIPE%20BOTACIN.pdf?sequence=1&isAllowed=y
14. Cybersecurity Career Guide: How to Land the Best Jobs - Blog – eForensics, dernier accès : février 12, 2026, https://eforensics3.rssing.com/chan-23437616/all_p12.html
15. Volatility, my own cheatsheet (Part 2): Processes and DLLs | Andrea Fortuna, dernier accès : février 12, 2026, https://andreafortuna.org/2017/07/03/volatility-my-own-cheatsheet-part-2-processes-and-dlls/
16. Command Reference · volatilityfoundation/volatility Wiki - GitHub, dernier accès : février 12, 2026, https://github.com/volatilityfoundation/volatility/wiki/command-reference
17. MemLabs: An Introduction To Memory Forensics - BananaMafia, dernier accès : février 12, 2026, https://bananamafia.dev/post/mem/
18. magazine - Microsoft Download Center, dernier accès : février 12, 2026, https://download.microsoft.com/download/2/6/e/26ef0117-c634-456c-878e-166c98c7c266/mdn_1409dg.pdf
19. Rekall Memory Forensic Framework - Googleapis.com, dernier accès : février 12, 2026, https://storage.googleapis.com/web.rekall-innovations.com/docs/Manual/Plugins/Windows/index.html?v=13
20. Plugin Reference — Rekall Forensics 1.7.2 documentation, dernier accès : février 12, 2026, https://rekall.readthedocs.io/en/latest/plugins.html