

ChocoShell: технічний аналіз

Аналітик	Turki Abdullah Almuraykhi
Період аналізу	28 вересня — 5 жовтня 2026 року
Середовище дослідження	Ізольована віртуальна машина під керуванням Windows; тимчасові локальні облікові записи
Основний зразок	Скрипт PowerShell, 55 877 байтів, 970 рядків
SHA-256	BE99857449D2856DD5A84E21C8A3D5E0E01456ADB440 62DDEC5A6B4970D8D42C
Обсяг аналізу	Поведінка ChocoShell, можливості отримання облікових даних, підвищення привілеїв, поетапне завантаження компонентів через інфраструктуру командування та керування (C2), механізми ексфільтрації даних і можливості виявлення.

Примітка щодо безпеки: Під час дослідження навмисних з'єднань із реальною інфраструктурою командування та керування (C2) не встановлювали. Динамічний аналіз проводили в ізольованому мережевому середовищі з окремим блокуванням з'єднань із C2.



1. Резюме дослідження

ChocoShell — компонент для викрадення облікових даних, написаний мовою PowerShell, який Microsoft Threat Intelligence пов'язує з кампанією CaptiveCrunch. Microsoft публічно ідентифікує зразок із тим самим хешем SHA-256, що досліджується в цьому звіті, як ChocoShell.

Компанія також визначає IP-адресу 213.145.86.112 як частину інфраструктури командування та керування (C2) ChocoShell. У цьому звіті атрибуція кампанії розглядається як зовнішній контекст кіберрозвідки (CTI): аналітик не проводив незалежної атрибуції оператора.

Статичний аналіз виявив багатоступеневу архітектуру, що передбачає підвищення привілеїв, обхід захисних механізмів, збирання даних браузерів і матеріалів активних сеансів, артефактів Microsoft TokenBroker, облікових даних Wi-Fi, збір інформації про систему та ексфільтрацію стиснених даних. Досліджуваний скрипт містить як механізми непомітного підвищення привілеїв до рівня локального адміністратора, так і резервний механізм RunAs, що передбачає видимий запит на підвищення привілеїв. Крім того, скрипт містить окремий механізм використання Chrome DevTools Protocol, який дає змогу запитувати значення файлів cookie у відкритому вигляді з локально запущеного браузера.

У ході контрольованого динамічного аналізу було встановлено два підтверджені сценарії поведінки. Під час першого запуску (Run 1), виконаного від імені звичайного користувача, з'явився запит контролю облікових записів Windows (UAC), який було відхилено. Другий запуск (Run 2B), виконаний від імені локального адміністратора з розділеним токеном доступу (split-token local administrator), виявив помилку в механізмі визначення адміністративних прав. Windows позначала ідентифікатор безпеки (SID) групи Administrators як deny-only, тоді як перевірка .NET Groups.Value, яку використовував зразок, повертала значення false. Унаслідок цього скрипт помилково визначив обліковий запис як такий, що не має адміністративних прав, обрав гілку RunAs і запустив процес PowerShell із високим рівнем цілісності (High integrity) через механізм Windows UAC/AppInfo.

Процес із підвищеними привілеями виконав вбудований закодований фрагмент коду (encoded stub) і завершив роботу протягом тієї самої секунди. Не було зафіксовано ані виконання наступного блоку ScriptBlock, отриманого через C2, ані повторного запуску браузера, звернення до TokenBroker, запуску процесу збирання облікових даних Wi-Fi чи зареєстрованого засобами аудиту читання файлів облікових даних браузера, для яких було налаштовано моніторинг.

Найімовірніше, ізольоване середовище перешкодило завантаженню наступного етапу шкідливого коду. Однак точний виняток WinHTTP не було зафіксовано, тому це пояснення залишається припущенням, а не підтвердженим фактом.



Аспект аналізу	Результати
Ідентифікація зразка	Хеш досліджуваного зразка повністю збігається з індикатором компрометації (IOC) ChocoShell, опублікованим Microsoft.
Підвищення привілеїв	Підтверджено динамічним аналізом: PowerShell із середнім рівнем цілісності (Medium) → UAC/AppInfo → PowerShell із високим рівнем цілісності (High) → виконання закодованого фрагмента коду.
Можливості отримання облікових даних	Підтверджено аналізом вихідного коду: бази даних і ключі браузерів, файли cookie через CDP, файли TokenBroker .tbres, профілі Wi-Fi.
Викрадення облікових даних під час Run 2B	Не спостерігалось. Для файлів облікових даних, за якими здійснювався моніторинг, не зафіксовано жодної відповідної події безпеки Windows із кодом 4663.
Поетапне завантаження через C2	Наявність URL-адрес C2 та логіки WinHTTP підтверджено аналізом вихідного коду. Виклик закодованого фрагмента коду з підвищеними привілеями також підтверджено. Успішне отримання наступного етапу коду з віддаленого сервера не підтверджено.
Викрадення облікових даних під час Run 2B	Аналіз вихідного коду підтверджує таку послідовність обробки даних: JSON → UTF-8 → GZIP → Base64 → {"gz":"..."} → POST-запит до /t/event.

2. Методологія аналізу та принципи оцінювання доказів

У цьому звіті розмежовано можливості, встановлені за результатами аналізу вихідного коду, поведінку, зафіксовану під час лабораторних досліджень, інформацію із зовнішніх джерел та аналітичні висновки. Рівні впевненості застосовуються до аналітичних оцінок і припущень, а не до безпосередньо встановлених фактів чи спостережень.

Позначення	Значення у звіті
ФАКТ (FACT)	Інформація, безпосередньо зафіксована в досліджуваному зразку або наведеному публічному джерелі.
СПОСТЕРЕЖЕННЯ (OBSERVATION)	Поведінка, безпосередньо зафіксована засобами моніторингу в контрольованому лабораторному середовищі.
ПІДТВЕРДЖЕНИЙ ВИСНОВОК (CONFIRMED FINDING)	Висновок, підтверджений кількома узгодженими прямими доказами, наприклад вихідним кодом і телеметрією процесів або подій.

ОБҐРУНТОВАНЕ ТВЕРДЖЕННЯ (SUPPORTED CLAIM)	Твердження, яке є правдоподібним і має суттєве доказове підґрунтя, але не отримало повного незалежного підтвердження.
АНАЛІТИЧНЕ ПРИПУЩЕННЯ (INFERENCE)	Інтерпретація аналітика, що ґрунтується на наявних доказах. Рівень впевненості зазначається як НИЗЬКИЙ / СЕРЕДНІЙ / ВИСОКИЙ (LOW / MEDIUM / HIGH).
НЕПЕРЕВІРЕНЕ ТВЕРДЖЕННЯ (UNVERIFIED CLAIM)	Твердження, наведене в коментарях до коду або іншому джерелі, яке не було перевірено незалежно.

2.1 Надійність джерел

Джерело	Роль	Оцінка надійності
Досліджуваний зразок PowerShell	Основний технічний артефакт	Безпосередній доказ наявності реалізованих у коді механізмів і вбудованих значень, але не підтверджує виконання кожної гілки коду.
Журнали Windows, Sysmon, PowerShell і Security	Основні джерела даних динамічного аналізу	Безпосередні докази подій, що відбулися протягом періоду виконання, охопленого моніторингом. Інтерпретація обмежується зафіксованими телеметричними даними.
Microsoft Threat Intelligence	Зовнішнє першоджерело кіберрозвідувальних даних (CTI)	Використовується для атрибуції кампанії на основі відкритих даних і підтвердження опублікованих індикаторів компрометації (IOC) ChocoShell.
Коментарі розробника у зразку	Твердження та контекст, що містяться в коді	Інформація з коментарів не вважається достовірною лише на підставі її наявності в коді. За можливості такі твердження перевіряються за допомогою незалежних джерел.

3. Ідентифікація зразка та контекст відкритих даних кіберрозвідки (СТІ)

Характеристика	Значення
Тип файлу	Скрипт PowerShell
Розмір	55 877 байтів
Кількість рядків	970
SHA-256	BE99857449D2856DD5A84E21C8A3D5E0E01456ADB44062DDEC5A6B4970D8D42C
SHA-256 декодованого аналітиком фрагмента коду з підвищеними привілеями	A133565A912C59D3D8FE577D44C574B73B5B4363A5BEFFCB488423F05375865F

Контекст зовнішньої кіберрозвідки. Microsoft Threat Intelligence ідентифікує зразок із тим самим хешем SHA-256 як ChocoShell і описує його як інструмент викрадення інформації (infostealer), написаний мовою PowerShell, який використовувався в кампанії CaptiveCrunch. Microsoft пов'язує кампанію CaptiveCrunch із групою Storm-2945, яку вважає окремим операційним підкласером Midnight Blizzard, а також визначає IP-адресу 213.145.86.112 як сервер командування та керування (C2) ChocoShell. Microsoft відзначає схожість тактик, технік і процедур (TTP) із діяльністю Forest Blizzard, проте зберігає атрибуцію кампанії CaptiveCrunch до Storm-2945. [1]

Межі атрибуції. У цьому звіті не проводиться незалежна атрибуція досліджуваного зразка державному суб'єкту. Зв'язки між шкідливим кодом і кампанією, а також між кампанією та її операторами ґрунтуються на оцінках зовнішніх джерел, на які наведено посилання. Подальші висновки аналітика зосереджені на поведінці та технічній реалізації конкретного зразка, ідентифікованого за його хешем.

4. Архітектура та етапи виконання

ChocoShell — це багатоступеневий інструмент для викрадення облікових даних, реалізований мовою PowerShell, а не скрипт вузького призначення. Статичний аналіз показав, що основний скрипт забезпечує початковий обмін інформацією про стан із сервером командування та керування (C2), налаштування механізмів обходу захисту, визначення необхідності підвищення привілеїв, збирання облікових даних, підготовку зібраної інформації до передавання, а також отримання додаткових компонентів та інструментів.

Вбудований допоміжний код для підвищення привілеїв (elevation stub) декодується з Base64, після чого повторно кодується у Base64 із використанням UTF-16LE. За потреби підвищення привілеїв цей код запускається через PowerShell із параметром -EncodedCommand.

Основний скрипт PowerShell

- > Перевірка середовища виконання та рівня привілеїв
- > Визначення необхідності підвищення привілеїв
- > Запуск допоміжного коду з підвищеними привілеями через -EncodedCommand
- > Надсилання повідомлення про стан на сервер C2
- > Завантаження наступного етапу коду та додаткових інструментів
- > Збирання облікових даних і даних сеансів
- > Стиснення та ексфільтрація зібраних даних

ФАКТ: Основний зразок містить жорстко задану базову URL-адресу `https://213.145.86.112` і використовує `WinHttp.WinHttpRequest.5.1` для мережевої взаємодії. Параметру `WinHTTP Option(4)` присвоєно значення `13056`, що налаштовує ігнорування помилок SSL.

ФАКТ: Вбудований допоміжний код, що виконується з підвищеними привілеями, містить запит для передавання інформації про стан на адресу `/t/pixel.gif?m=stub_start` і шлях `/cdn/chunks/vendor-9f3a1c.min.js?stub=1` для завантаження наступного етапу коду. Після завантаження отриманий код виконується безпосередньо в оперативній пам'яті.

ФАКТ: Основний етап також використовує шлях `/cdn/chunks/polyfill-7e2b.min.js` для отримання додаткових інструментів і `/t/event` для передавання даних.

5. Обхід захисних механізмів і протидія аналізу

5.1 Маніпулювання станом AMSI

Досліджуваний зразок містить код, який використовує механізм рефлексії (reflection) для доступу до `System.Management.Automation.AmsiUtils` та внутрішнього поля `amsiInitFailed`. Аналогічний механізм реалізовано в допоміжному коді, що виконується з підвищеними привілеями. Це свідчить про спробу змінити роботу AMSI до виконання наступних компонентів PowerShell. Водночас результати аналізу не дають підстав стверджувати, що AMSI вдалося вимкнути в лабораторному середовищі, оскільки незалежний моніторинг його внутрішнього стану не проводився.

5.2 Кодування та виконання наступних етапів у пам'яті

Допоміжний код для підвищення привілеїв зберігається у вигляді ASCII-тексту, закодованого в Base64. Під час виконання він декодується, після чого повторно кодується у Base64 з використанням UTF-16LE для запуску через параметр PowerShell - `EncodedCommand`. Наступні компоненти передбачено декодувати та виконувати безпосередньо в оперативній пам'яті у вигляді блоку сценарію (ScriptBlock). Такий підхід зменшує залежність від традиційних виконуваних файлів і ускладнює виявлення засобами захисту, що ґрунтуються на перевірці файлів.

5.3 Перевірка віртуального середовища за часом виконання

Основний скрипт містить перевірку часу виконання, що передбачає затримку тривалістю 2,5 секунди. Якщо фактично виміряна затримка виявляється меншою за встановлене порогове значення, скрипт може достроково завершити роботу. Наявність цього механізму у вихідному коді свідчить про реалізовану можливість протидії аналізу. Під час аналізу значення змінної, у якій зберігався виміряний час, окремо не відстежували. Тому неможливо підтвердити, чи спрацювала перевірка віртуального середовища під час будь-якого з двох коректно проведених запусків.

6. Підвищення привілеїв

Підвищення привілеїв відіграє ключову роль у роботі ChocoShell, оскільки для реалізації низки його функцій необхідні підвищені привілеї або рівень доступу, еквівалентний обліковому запису SYSTEM. У вихідному коді реалізовано різні сценарії виконання для користувачів з адміністративними правами та без них.

Сценарій виконання	Реалізований механізм	Результати динамічного аналізу
Локальний адміністратор із розділеним токеном доступу (split-token)	Обхід UAC через SilentCleanup із підміною змінної середовища windir; підміна реєстрації класу AppX через wsreset.exe; підміна обробника папок оболонки Windows через sdclt.exe / KickOffElev.	Механізми наявні у вихідному коді, але їх виконання під час Run 2B не зафіксовано, оскільки перевірка належності до групи локальних адміністраторів у цьому середовищі повернула значення false.
Користувач без адміністративних прав / резервний сценарій	Запуск PowerShell за допомогою Start-Process powershell -Verb RunAs із передаванням закодованого допоміжного скрипту через параметр - EncodedCommand. Скрипт запускається у прихованому неінтерактивному режимі.	Механізм спостерігався під час Run 1 і Run 2B. У Run 1 запит UAC було відхилено. У Run 2B створено дочірній процес PowerShell із високим рівнем цілісності (High integrity).

6.1 Помилка визначення прав локального адміністратора з розділеним токеном доступу

Під час Run 2B було виявлено помилку в реалізації перевірки прав локального адміністратора, яка проявилася в досліджуваному середовищі. Обліковий запис ChocoAdmin належав до групи BUILTIN\Administrators, а команда whoami /groups відображала ідентифікатор безпеки (SID) S-1-5-32-544 зі статусом «Group used for deny only» (група, що використовується лише для відмови в доступі) у токені із середнім рівнем цілісності (Medium integrity).

Водночас перевірка за допомогою виразу .NET, який використовувався у зразку, — WindowsIdentity.GetCurrent().Groups.Value -contains S-1-5-32-544 — повертала значення false. Метод IsInRole(Administrator) також повертав false, що є очікуваною поведінкою для обмеженого токена із середнім рівнем цілісності.

ПІДТВЕРДЖЕНИЙ ВИСНОВОК: Оскільки обидві перевірки повернули false, ChocoShell помилково визначив локального адміністратора з розділеним токеном доступу як користувача без адміністративних прав. Унаслідок цього скрипт обрав сценарій підвищення привілеїв через RunAs/UAC замість передбачених для локальних адміністраторів механізмів непомітного підвищення привілеїв.

7. Збирання облікових даних і даних сеансів

Отримання облікових даних є однією з найважливіших можливостей, реалізованих у досліджуваному коді. Скрипт передбачає відновлення ключів шифрування браузерів, збирання файлів баз даних браузерів у вихідному вигляді, отримання значень файлів cookie у відкритому вигляді через Chrome DevTools Protocol (CDP), збирання артефактів TokenBroker та вилучення даних із профілів Wi-Fi.

Важливо розмежовувати можливості, підтверджені аналізом вихідного коду, та фактично зафіксовану поведінку: хоча всі перелічені механізми реалізовані у скрипті, під час Run 2B виконання не дійшло до етапу збирання даних.

7.1 Обробка файлу Local State та ключів шифрування Chromium

Функція Derive-MasterKey зчитує файл Local State браузера та спочатку перевіряє наявність параметра `os_crypt.app_bound_encrypted_key`. Значення цього параметра декодується з Base64, після чого перевіряється префікс APPB. За потреби захищений блок даних передається механізму розшифрування, що працює в контексті облікового запису SYSTEM. Для цього у вихідному коді передбачено використання функції `Invoke-Impersonate-System`, яка дає змогу тимчасово виконувати операції з токеном доступу SYSTEM, із подальшим викликом `ProtectedData.Unprotect`. Після цього скрипт повертається до початкового контексту безпеки та намагається зняти другий рівень захисту DPAPI в контексті користувача.

Для 32-байтових даних, характерних для Edge, Brave, Opera та Vivaldi, скрипт може повертати отримане значення як ключ шифрування ABE (Application-Bound Encryption), а також, за потреби, відновлювати ключ `encrypted_key`, що використовувався в попередніх версіях механізму шифрування, як резервний варіант через DPAPI.

Для Chrome у сценарії з прапорцем `flag-3` вихідний код передбачає використання функції `Parse-ChromeKeyBlob`, розшифрування засобами NCrypt та обробку даних за алгоритмом AES-GCM. Основний скрипт посилається на кілька допоміжних функцій, визначення яких відсутні в досліджуваному файлі. Перед використанням відповідних механізмів передбачено завантаження додаткових інструментів.

7.2 Файли cookie та дані авторизації

Для кожного профілю Chromium функція збирання даних знаходить базу даних `Network\Cookies` і копіює її до тимчасового файлу. Отриманий запис містить вихідну базу даних cookie, закодовану в Base64, а також відновлений ключовий матеріал і тип ключа.

За наявності бази даних Login Data скрипт також копіює її та кодує в Base64, але лише за умови успішного виконання операцій із базою даних Cookies для відповідного профілю.

ВАЖЛИВЕ ОБМЕЖЕННЯ: На досліджуваному етапі не виявлено механізмів SQL-обробки окремих записів у полях `password_value` або `encrypted_value` баз даних Chromium. Натомість скрипт готує до передавання базу даних Login Data у вихідному вигляді разом із ключовим матеріалом. У цьому звіті не встановлюється, на якому з наступних етапів відбувається вилучення паролів з окремих записів бази даних.

7.3 Отримання файлів cookie через Chrome DevTools Protocol

Окремий механізм передбачає запуск підтримуваних браузерів із відкритим портом віддаленого налагодження та використання Chrome DevTools Protocol (CDP). У вихідному коді жорстко задано такі порти: 19876 для Chrome, 19877 для Edge та 19878 для Brave.

Скрипт отримує інформацію про доступні цілі налагодження через /json, встановлює з'єднання WebSocket і надсилає команду Network.getAllCookies. Отримані записи cookie містять поля host, name, value, path, expiration, Secure, HttpOnly і SameSite.

ФАКТ: Цей механізм призначений для отримання значень cookie у відкритому вигляді безпосередньо з браузера, тому не потребує окремого розшифрування кожного запису cookie поза браузером. Microsoft також описує аналогічний механізм використання CDP у своєму публічному аналізі ChocoShell. [1]

7.4 Браузери сімейства Firefox

Скрипт передбачає роботу зі структурами профілів Firefox, Waterfox, LibreWolf, Floorp і Zen. Він копіює файл cookies.sqlite та зберігає вміст бази даних у вихідному вигляді, закодованому в Base64. На цьому етапі ChocoShell копіює бази даних SQLite без аналізу окремих записів.

7.5 Microsoft TokenBroker

Скрипт здійснює пошук файлів *.tbres у каталозі %LOCALAPPDATA%\Microsoft\TokenBroker\Cache та зберігає в об'єкті результатів назву кожного файлу, його розмір, вміст у вихідному вигляді, закодований у Base64, і час останньої зміни.

Під час лабораторного аналізу вміст окремих файлів .tbres не досліджували на рівні їхнього змістового наповнення. Отже, підтвердженою можливістю є саме збирання файлів TokenBroker у вихідному вигляді.

У публічному аналізі ChocoShell Microsoft зазначає, що метою операторів є викрадення даних сеансів Microsoft 365 та систем керування цифровою ідентичністю. [1]

7.6 Профілі Wi-Fi

Скрипт виконує команду netsh wlan show profiles, отримує назви збережених профілів Wi-Fi та запитує інформацію про кожен із них із параметром key=clear.

Після цього він знаходить значення поля Key Content і зберігає в об'єкті результатів пари SSID і пароль. Під час Run 2B виконання не дійшло до етапу збирання цих даних.

8. Командування та керування та ексфільтрація даних

Артефакт	Призначення	Статус підтвердження
213.145.86.112	HTTPS-сервер C2 із заданою в коді адресою	Наявний у зразку; Microsoft незалежно ідентифікує його як C2-сервер ChocoShell. [1]
/t/pixel.gif?m=<status>	Шлях для повідомлень про стан (beacon)	Наявний у зразку; цей механізм підтверджується публічними матеріалами Microsoft. [1]
/cdn/chunks/polyfill-7e2b.min.js	Завантаження додаткових інструментів	Наявний у зразку; цей шлях підтверджується публічними матеріалами Microsoft. [1]
/cdn/chunks/vendor-9f3a1c.min.js?stub=1	Завантаження наступного етапу коду з підвищеними привілеями	Наявний у допоміжному коді, декодованому аналітиком; фактичне отримання вмісту незалежно не підтверджено.
/t/event	Кінцева точка ексфільтрації	Наявна у зразку; цей шлях підтверджується публічними матеріалами Microsoft. [1]

Структура результатів містить категорії browsers, firefox, cdp_cookies, tokens, wifi та system. На завершальному етапі зібрані дані перетворюються на компактний JSON, кодуються в UTF-8, стискаються за допомогою GZIP, після чого стиснені дані кодуються в Base64, поміщаються в JSON-об'єкт і передаються через POST-запит WinHTTP на адресу /t/event.

- Об'єкт результатів**
- > ConvertTo-Json -Compress
 - > Байти UTF-8
 - > GZIP
 - > Base64
 - > {"gz": "<base64>"}
 - > WinHTTP POST /t/event

НЕПЕРЕВІРЕНЕ ТВЕРДЖЕННЯ: У коментарях до вихідного коду згадуються обмеження nginx/VPS щодо розміру тіла запиту та деталі реалізації серверної частини. Ці твердження не були незалежно перевірені, тому їх не слід вважати підтвердженими характеристиками інфраструктури.

9. Контрольований динамічний аналіз

9.1 Заходи безпеки

- Тимчасова віртуальна машина Windows зі знімками стану до та після етапів тестування.
- Відсутність маршруту за замовчуванням для IPv4 під час запусків із моніторингом; перед Run 2B також видалено маршрут за замовчуванням для IPv6.
- Окреме правило брандмауера для блокування вихідних з'єднань із 213.145.86.112.
- Жодних навмисних підключень, ping-запитів або перевірок доступності реального сервера C2.
- Під час Run 2B використовувалися Sysmon, PowerShell Script Block Logging, аудит створення процесів, аудит файлової системи та налаштування SACL для реєстрації подій Security 4663 щодо вибраних файлів браузерів і TokenBroker.

9.2 Run 1 — Звичайний користувач / запит UAC відхилено

Параметр	Зафіксоване значення
Обліковий запис	ChocoLab — звичайний користувач
PID основного процесу	7488
Рівень цілісності	Medium
Підвищення привілеїв	Видимий запит UAC/RunAs; відхилений аналітиком
Доступ до облікових даних	Не зафіксовано пов'язаної з PID 7488 активності щодо TokenBroker, облікових даних браузерів або CDP
Мережева активність	У проаналізованих даних Procmon не зафіксовано TCP/UDP-активності, пов'язаної з PID 7488
Результат	Основний процес завершився; ланцюжок виконання з підвищеними привілеями не запустився

9.3 Run 2A — Некоректний тест, виключений із результатів аналізу

Проміжний тест було збережено для документування, але виключено з результатів аналізу поведінки. Sysmon зафіксував командний рядок запущеного процесу PowerShell як -File "", що свідчить про те, що змінна зі шляхом до зразка не була визначена в цьому сеансі PowerShell. Унаслідок цього зразок не виконався, тому результати цього запуску не використовуються як докази поведінки шкідливого коду.



9.4 Run 2B — Локальний адміністратор із розділеним токеном доступу

Час (місцевий)	PID	Рівень цілісності	Зафіксована подія
16:06:13	8432	Medium	Початковий зразок ChocoShell запущено від імені ChocoAdmin.
16:06:14	2492	SYSTEM	Процес consent.exe створено службою AppInfo через svchost.exe, що відповідає процедурі підвищення привілеїв через UAC.
16:06:19	2952	High	Дочірній процес PowerShell, створений процесом із PID 8432, запущено з параметрами - NonI -W H -EncodedCommand.
16:06:19	2952	High	Подія PowerShell 4104 зафіксувала декодований вміст команди допоміжного коду, що виконувався з підвищеними привілеями.
16:06:19	2952	High	Процес завершився протягом тієї самої секунди.

ПІДТВЕРДЖЕНИЙ ВИСНОВОК: Під час Run 2B зафіксовано підвищення привілеїв із рівня Medium до High через механізм Windows RunAs/UAC/AppInfo та запуск закодованого допоміжного коду ChocoShell для підвищення привілеїв із рівнем цілісності High.

СПОСТЕРЕЖЕННЯ: Пошук подій безпеки 4663 щодо об'єктів Local State, Login Data, Network\Cookies, TokenBroker і .tbres, для яких було налаштовано аудит, не виявив жодної відповідної події доступу до облікових даних протягом виконання Run 2B.

СПОСТЕРЕЖЕННЯ: Протягом Run 2B не зафіксовано процесів msedge, chrome, brave, netsh, taskkill, vssadmin або wmic, пов'язаних з етапом збирання даних. Після виконання допоміжного коду з підвищеними привілеями також не зафіксовано жодного наступного блоку ScriptBlock, отриманого через C2.

АНАЛІТИЧНЕ ПРИПУЩЕННЯ — ВИСОКИЙ РІВЕНЬ ВПЕВНЕНOSTІ: Найімовірніше, виконання допоміжного коду з підвищеними привілеями завершилося під час спроби завантажити наступний етап через WinHTTP через необроблений виняток. Це узгоджується з мережевою ізоляцією лабораторного середовища, обмеженнями брандмауера та активністю PowerShell, пов'язаною з форматуванням помилки, одразу після виконання допоміжного коду. Оскільки точний текст винятку не було зафіксовано, причина зупинки залишається припущенням, а не підтвердженим висновком.

10. Оцінка результатів аналізу

ID	Класифікація	Висновок	Докази
CF-01	ПІДТВЕРДЖЕНИЙ ВИСНОВОК	SHA-256 досліджуваного зразка збігається з хешем ChocoShell, опублікованим Microsoft.	Хеш зразка + таблиця IOC Microsoft [1]
CF-02	ПІДТВЕРДЖЕНИЙ ВИСНОВОК	У зразку реалізовано механізм підвищення привілеїв через PowerShell RunAs, виконання якого зафіксовано під час динамічного аналізу.	Відповідна гілка вихідного коду + телеметрія процесів Run 1/Run 2B
CF-03	ПІДТВЕРДЖЕНИЙ ВИСНОВОК	Під час Run 2B після обробки запиту UAC/AppInfo було запущено закодований допоміжний код ChocoShell із рівнем цілісності High.	Ланцюжок процесів Sysmon + подія PowerShell 4104
CF-04	ПІДТВЕРДЖЕНИЙ ВИСНОВОК	У досліджуваному коді реалізовано збирання даних браузерів, облікових даних і даних сеансів, а також ексфільтрацію стиснених даних.	Безпосередній аналіз вихідного коду
SC-01	ОБҐРУНТОВАНЕ ТВЕРДЖЕННЯ	Під час Run 2B виконання не дійшло до етапу збирання облікових даних, який можна було б зафіксувати.	0 відповідних подій 4663 + відсутність дочірніх процесів, пов'язаних зі збиранням даних
IN-01	АНАЛІТИЧНЕ ПРИПУЩЕННЯ — ВИСОКИЙ РІВЕНЬ ВПЕВНЕНOSTІ	Найімовірніше, виконання допоміжного коду з підвищеними привілеями завершилося під час заблокованого завантаження наступного етапу через C2.	Негайне завершення процесу + мережева ізоляція + блоки сценарію, пов'язані з форматуванням помилок; точний текст винятку відсутній

11. Можливості виявлення та пошуку загроз

Кореляція поведінкових ознак забезпечує надійніші можливості виявлення, ніж окремі текстові рядки. Наведені нижче підходи ґрунтуються на аналізі вихідного коду та лабораторній телеметрії. Перед використанням у робочому середовищі для формування сповіщень механізми, наявність яких підтверджено лише аналізом коду, слід перевірити в цільовому середовищі.

Поведінка	Орієнтири для виявлення	Статус
Підвищення привілеїв PowerShell	Процес powershell.exe із рівнем цілісності Medium запускає powershell.exe із параметром -Verb RunAs або дочірній процес із підвищеними привілеями, що містить -EncodedCommand.	Спостерігалось
Послідовність UAC/ AppInfo	Запуск consent.exe службою AppInfo поблизу за часом до підозрілого процесу PowerShell із подальшим запуском PowerShell із рівнем цілісності High.	Спостерігалось
Патерн ChocoShell C2	Звернення PowerShell/WinHTTP до 213.145.86.112 із використанням /t/pixel.gif? m= або /t/event.	Вихідний код + зовнішні дані СТІ
Гілка SilentCleanup	Зміна HKCU\Environment\windir із подальшим виконанням DiskCleanup\SilentCleanup та видаленням значення.	Лише потенційна можливість у цьому лабораторному середовищі
Отримання cookie через CDP	Запуск браузера з параметром --remote-debugging-port=19876/19877/19878 із подальшим доступом до CDP/WebSocket через localhost.	Лише потенційна можливість у цьому лабораторному середовищі
Читання файлів облікових даних	Читання PowerShell файлів браузерів Local State, Network\Cookies, Login Data або TokenBroker\Cache*.tbres.	Потенційна можливість; моніторинг налаштовано, але під час Run 2B не спостерігалось

11.1 Артефакти виявлення, розроблені аналітиком

Артефакт	Статус перевірки
ChocoShell_v1.yar	Логіку перевірено аналітиком; 0 хибнопозитивних спрацювань серед 258 легітимних файлів PowerShell і модулів у тестовому наборі аналітика. Перевірка за допомогою окремого рушія YARA під час задокументованого тестування була недоступна.
ChocoShell_CDP_Sigma_v1.yml	Чернетка поведінкового правила Sigma; під час задокументованого тестування перевірка парсером не проводилася.
Кореляція Defender KQL / Splunk	Логіка пошуку загроз, розроблена аналітиком; до перевірки на схемах телеметрії робочого середовища її слід вважати чернеткою.



12. Відповідність технікам MITRE ATT&CK

Наведені нижче відповідності охоплюють можливості, реалізовані в досліджуваному зразку, та поведінку, зафіксовану під час тестування. Позначення «Можливість» означає, що відповідний механізм наявний у коді, але не обов'язково виконувався під час контрольованого запуску.

Техніка	ID	Підстава	Статус
PowerShell	T1059.001	Основне виконання та поетапні команди PowerShell	Спостерігалось
Обфускація/стиснення файлів та інформації	T1027	Base64/EncodedCommand та закодований вміст етапів	Спостерігалось / можливість
Деобфускація/ декодування файлів або інформації	T1140	Декодування Base64 під час виконання та створення ScriptBlock	Можливість
Послаблення захисних механізмів	T1562.001	Маніпулювання станом AMSI; посилання на інструменти, пов'язані з Defender	Можливість
Обхід контролю облікових записів користувачів (UAC)	T1548.002	Гілки SilentCleanup/wsreset/sdclt для локальних адміністраторів	Можливість
Отримання облікових даних із веббраузерів	T1555.003	Бази даних браузерів Chromium/ Firefox та ключовий матеріал	Можливість
Викрадення файлів cookie вебсеансів	T1539	Механізм CDP Network.getAllCookies	Можливість
Маніпулювання токенами доступу: імітація/викрадення токенів	T1134.001	Посилання на допоміжну функцію імітації токена SYSTEM для механізму ABE	Можливість
Заплановане завдання/ робота: заплановане завдання	T1053.005	Запуск SilentCleanup і тимчасові завдання для запуску браузера	Можливість
Протокол прикладного рівня: вебпротоколи	T1071.001	HTTPS/WinHTTP C2	Можливість
Ексфільтрація через канал C2	T1041	POST /t/event	Можливість
Видалення файлів	T1070.004	Видалення тимчасових артефактів підвищення привілеїв і збирання даних	Можливість
Ухилення від виявлення на основі часу	T1497.003	Перевірка віртуальної машини/ пісочниці за різницею часу після затримки	Можливість

13. Реєстр доказів

Артефакт	SHA-256 / примітка
Оригінальний зразок	BE99857449D2856DD5A84E21C8A3D5E0E01456ADB44062DDEC5A6B4970D8D42C
Допоміжний код із підвищеними привілеями, декодований аналітиком	A133565A912C59D3D8FE577D44C574B73B5B4363A5BEFFCB488423F05375865F
Run 1 Procmon PML	D86C8F553BEE6471DB4F5717566A8A2D12433A48AA85A35F8F48AD783726E1BE
Run 1 — відфільтрований Procmon CSV	BEEBBDAFA2D21AFA4457A67F8505B9029EECE3A3EE8423EFB073DC68EE0CB4E8
Run 2 — конфігурація Sysmon	9BE0212D51AA1452BBD2A5CFD0A9B07E3243BFD56FC8E6B14ADA6DF230A8E940
Run 2B — лог Sysmon	run2B_sysmon.evtx — хеш не наведено у звіті, оскільки його не було зафіксовано в наданому протоколі доказів.
Run 2B — лог PowerShell	run2B_powershell.evtx — хеш не наведено у звіті, оскільки його не було зафіксовано в наданому протоколі доказів.
Run 2B — лог планувальника завдань	run2B_taskscheduler.evtx — хеш не наведено у звіті, оскільки його не було зафіксовано в наданому протоколі доказів.
Run 2B — лог безпеки	run2B_security.evtx — хеш не наведено у звіті, оскільки його не було зафіксовано в наданому протоколі доказів.
Run 2B — підсумкова оцінка	311D031EF9C496CAA9A4B595D87FF444E80D4949308949E26CC9B4C7F83F0DA8

14. Scope, Limitations, and What Was Not Proven

- Навмисних з'єднань із реальним сервером C2 не встановлювали. Лабораторне середовище було ізольоване, а IP-адресу виявленого сервера C2 окремо заблоковано.
- Наступний етап коду, який мав завантажуватися за шляхом /cdn/chunks/vendor-9f3a1c.min.js?stub=1, не було отримано з реальної інфраструктури.
- Аналіз вихідного коду переконливо підтверджує можливості викрадення облікових даних, однак під час Run 2B не було отримано динамічних доказів збирання даних браузерів, TokenBroker або Wi-Fi.
- Під час лабораторного аналізу вміст файлів TokenBroker .tbres не досліджували на семантичному рівні. Збирання цих файлів у вихідному вигляді підтверджено аналізом коду, тоді як семантику токенів описано в публічних матеріалах Microsoft.

- На досліджуваному етапі не виявлено обробки окремих записів паролів Chromium. База даних Login Data у вихідному вигляді та ключовий матеріал готуються для подальшого використання.
- У коді наявний механізм маніпулювання AMSI, проте успішне вимкнення AMSI не було незалежно перевірено засобами моніторингу.
- Точний текст винятку, який призвів до завершення процесу з PID 2952, не було зафіксовано. Тому висновок про невдале завантаження даних із мережі залишається аналітичним припущенням із високим рівнем впевненості.
- Атрибуція кампанії ґрунтується на публічних матеріалах Microsoft і не була незалежно встановлена аналітиком.

Висновки

Проаналізований зразок ChocoShell — це багатоступеневий викрадач облікових даних на базі PowerShell, розроблений із урахуванням підвищення привілеїв та обходу захисних механізмів. Вихідний код свідчить про значну увагу до сучасних механізмів захисту облікових даних у браузерях, зокрема до роботи з Application-Bound Encryption, виконання операцій у контексті SYSTEM, збирання необроблених баз даних браузерів та паралельного використання CDP для отримання значень файлів cookie у відкритому вигляді. Збирання даних TokenBroker і Wi-Fi розширює потенційні наслідки успішного зараження за межі викрадення браузерних паролів.

Динамічне тестування підтвердило роботу механізму підвищення привілеїв без встановлення з'єднання з реальним сервером C2. Найважливішим результатом лабораторного дослідження стало виявлення помилки розпізнавання адміністратора з розділеним токеном (split-token administrator): у протестованому середовищі Windows ChocoShell не зміг розпізнати SID групи Administrators з атрибутом deny-only через механізм переліку груп .NET і тому обрав шлях із видимим запитом RunAs. У результаті послідовність UAC/AppInfo призвела до запуску процесу PowerShell із високим рівнем цілісності (High integrity), який виконав закодований допоміжний код із підвищеними привілеями.

Під час лабораторного дослідження не було зафіксовано успішного завантаження наступного етапу або збирання облікових даних. Це не означає, що такі можливості відсутні, а лише визначає межі того, що вдалося підтвердити динамічним тестуванням. Тому підсумкова оцінка розмежовує реалізовані можливості викрадення облікових даних і фактично спостережуване виконання, не приписуючи зразку поведінку, яку не було зафіксовано.

Джерела

[1] Microsoft Threat Intelligence. «CaptiveCrunch: Midnight Blizzard атакує мандрівників у всьому світі з метою поширення шкідливого ПЗ та викрадення облікових даних». Microsoft Security Blog, 31 липня 2026 року.

Доказові матеріали аналітика: вихідний код зразка ChocoShell, дані Procmon/Sysmon із Run 1, журнали Sysmon/PowerShell/Security/Task Scheduler із Run 2B, декодований допоміжний код підвищення привілеїв та артефакт підсумкової оцінки Run 2B.