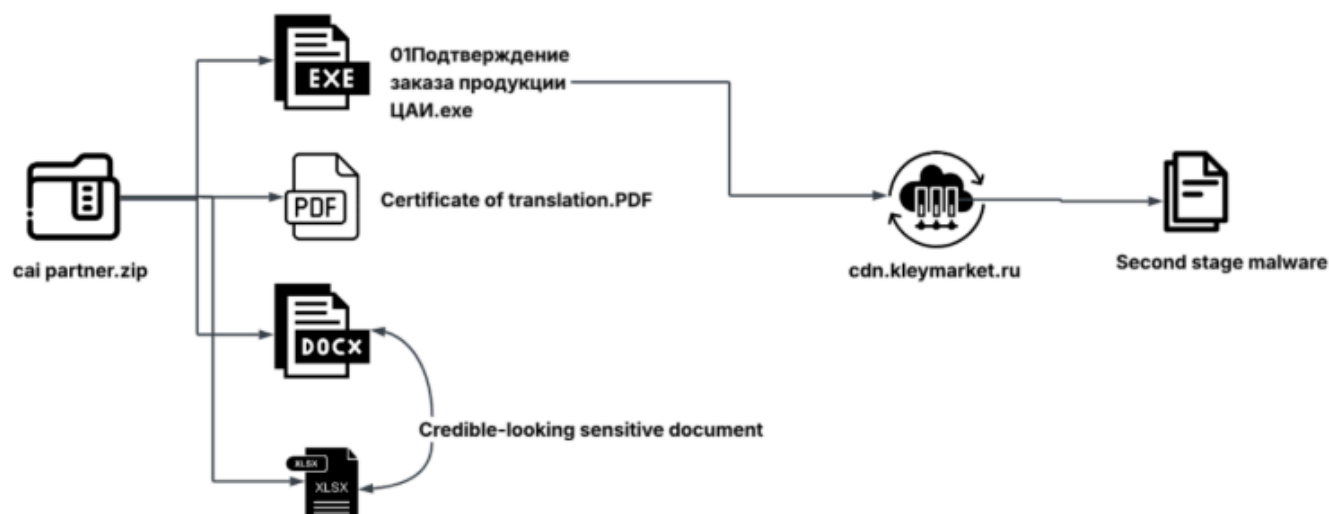


Rust-Based Malware Targets Eurasian Unmanned Aviation Sector Ahead of Moscow Summit

 seqrite.com/blog/operation-silent-rotor-rust-malware-unmanned-aviation-sector

Priya Patel

May 6, 2026



[Home](#) / [Malware](#) / Operation Silent Rotor: Targeted Campaign Compromises Unmanned Aviation Sector Ahead of Moscow Summit

Operation Silent Rotor: Targeted Campaign Compromises Unmanned Aviation Sector Ahead of Moscow Summit

Table of Content

- Introduction
- Key Targets
 - Industries Affected
 - Geographical focus
- Infection Chain
- Initial Findings
 - Looking into the Decoy Documents
- Technical Analysis
 - Stage 1 – Analysis of Malicious Executable
 - Stage 2 – Second stage payload dropper
- Infrastructure & Attribution
- Conclusion
- Seqrite Protection
- Indicators of Compromise (IOCs)
- MITRE ATT&CK Mappinga
- Authors

Introduction

SEQRITE Labs has been actively tracking emerging threats and recently identified a campaign targeting professionals and organizations in the Eurasian unmanned aviation sector. The campaign appears to be strategically timed to align with the XIII Eurasian International Forum “Unmanned Aviation 2026,” scheduled for April 23, 2026, in Moscow, a major international event focused on unmanned aerial systems (UAS).

The campaign distributes malicious suspected email attachments through spear phishing, delivering an archive named `cai partner.zip`. The archive contains a Rust-based executable disguised as a legitimate order confirmation document from the Russian Aeronautical Information Center (ЦАИ/CAI).

In the following sections, we will analyze the phishing lures used in this campaign, followed by a technical analysis of the malicious executable. We will also examine how the file downloads and executes a second-stage payload. Additionally, we will review the infrastructure and C2 communication used, and finally map the observed behavior to MITRE ATT&CK tactics and techniques.

Key Targets

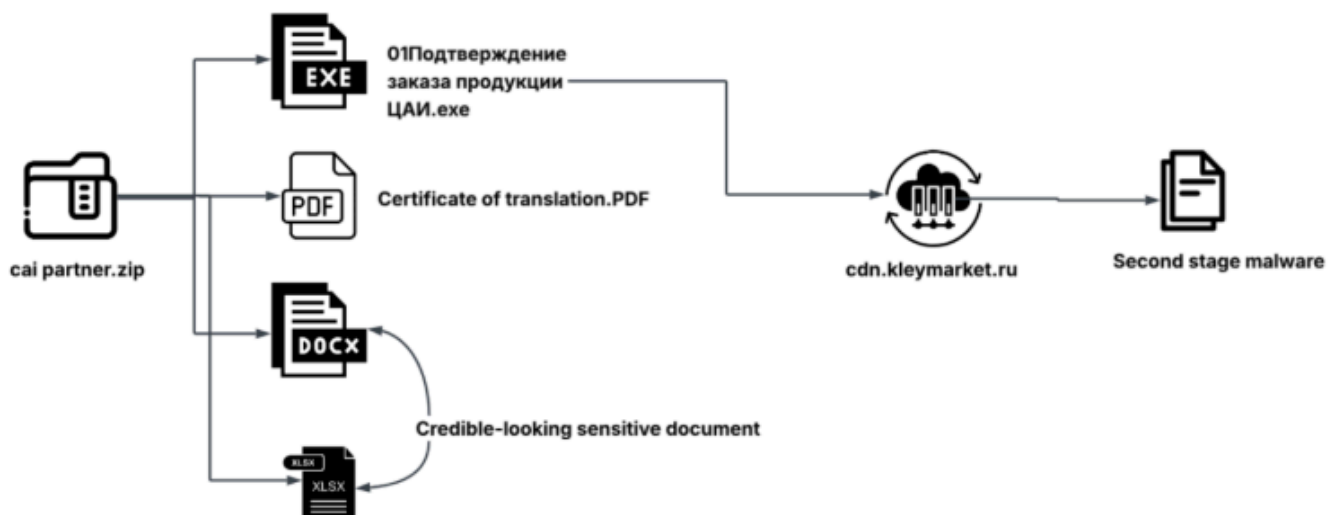
Industries Affected

- Unmanned Aviation Systems (UAS/UAV) sector
- Aeronautical information services

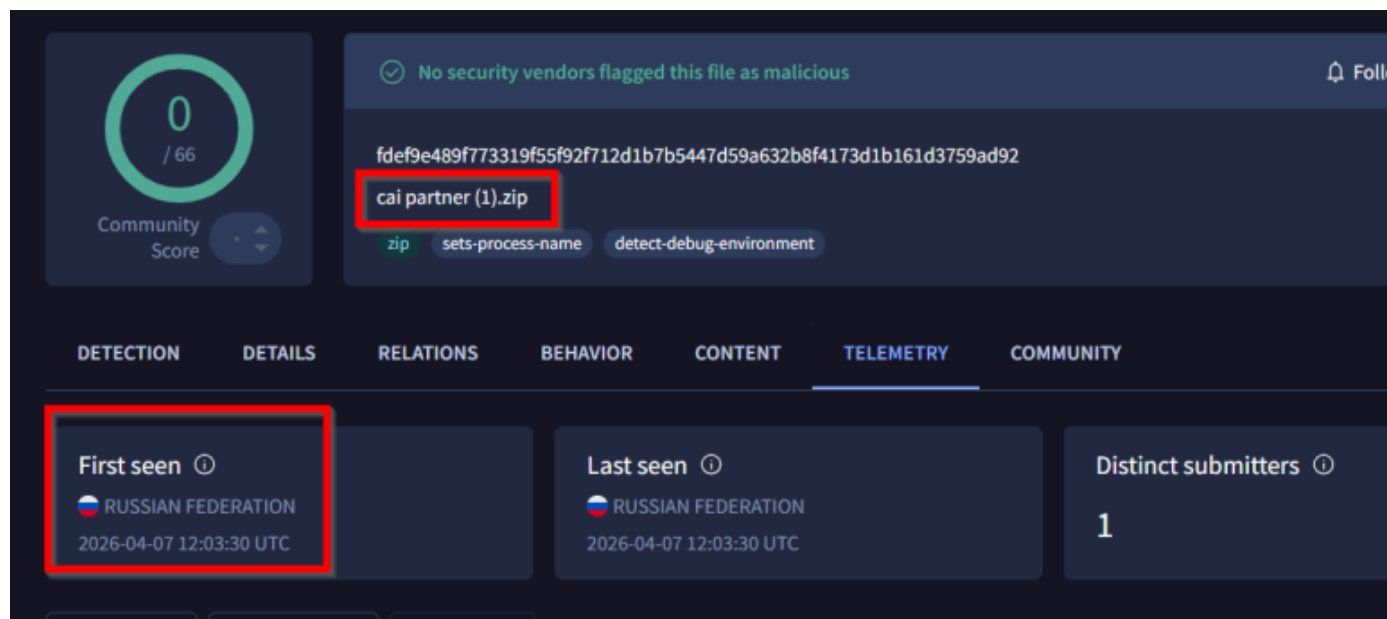
Geographical focus

- Russia
- Tajikistan
- Central Asia
- Middle East & Europe

Infection Chain



Initial Findings



As we were investigating recent ongoing campaigns, we identified a suspicious ZIP file named `cai partner (1).zip` on the public sandbox, VirusTotal. In the initial observation, we saw that the file was submitted from Russia and had zero detections during its first surface against our hunting rules. However, the file name and its contents appeared suspicious, which drew our attention.



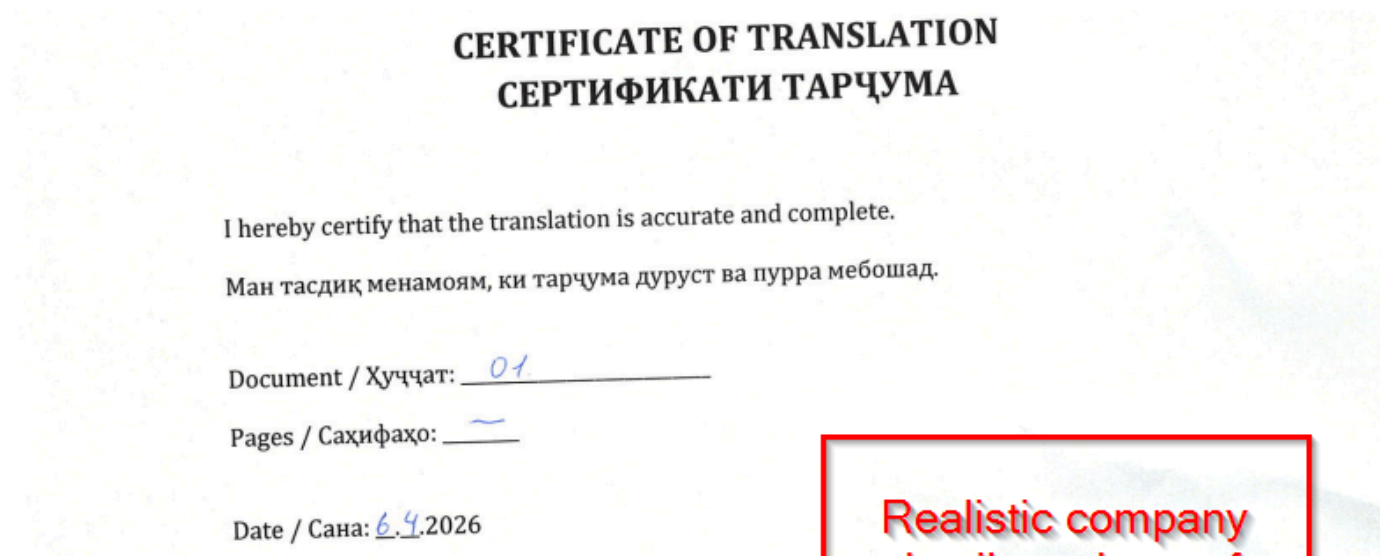
Further investigation led us to another archive named `cai partner.zip`, which serves as the primary carrier of both the lure documents and the malicious executable. Upon extracting the contents of this archive, we observed that it contains four files. One of these is a binary executable named “Подтверждение заказа продукции ЦАИ.exe”, which translates to “Confirmation of order of CAI products.exe”.

In addition to the executable, the archive includes multiple decoy documents such as a PDF, DOCX, and XLSX file, all designed to appear as legitimate business-related documents.

Looking into the Decoy Documents

PDF Lure Document

The first document that caught our attention is a PDF file extracted from the archive. The file is named Certificate of translation.PDF and is designed to appear as a legitimate certification document.



Upon looking into the document, it presents itself as an official translation certificate. It includes details such as document number, date, translator name, signature, and company information (SOMON TRANSLATIONS, Dushanbe, Tajikistan), along with stamps and seals to enhance its authenticity.

Therefore, by looking at the detailing of the document, including the use of multiple languages, formal structure, and official-looking stamps and signatures, it appears that the lure is designed to target professionals involved in international business or translation-related activities.

DOCX Lure Document

After analyzing the document, we found that this file, named Confirmation of CAICA products order and arrangement of a meeting at the Unmanned Aviation – 2026 forum to discuss payment details and sign a long-term contract.docx is similar to the document displayed to the victim immediately after the execution of the malicious executable.

We will analyze this document in detail in the next section, where we demonstrate how it is spawned by the executable during runtime.

XLSX Lure Document

The third document identified within the archive is an Excel file named summary_order_cai_final.xlsx

A	B	C	D	E	F	G	H	I
No.	Product	Quantity	Unit	Update Frequency	Remarks	No.	Product	Quantity
1	NDB for "Aerolotsia PRO" ? Worldwide	2	licenses	AIRAC (28 days)	For ground planning stations	1	NDB for "Aerolotsia PRO" ? Worldwide	2
2	NDB for "Aerolotsia PRO" ? EEU+EUR+MES	1	license	AIRAC (28 days)	For mobile workstation	2	NDB for "Aerolotsia PRO" ? EEU+EUR+MES	1
3	NOTAM database (EEU+EUR+MES zone)	12	licenses	hourly + AFTN	10 for onboard tablets, 2 for ground PCs	3	NOTAM database (EEU+EUR+MES zone)	12
4	Electronic AIP "Airport Optima"	12	licenses	AIRAC (28 days)	For tablets (iOS/Windows)	4	Electronic AIP "Airport Optima"	12
5	Electronic AIP "RUSSIA"	12	licenses	AIRAC (28 days)	For tablets (iOS/Windows)	5	Electronic AIP "RUSSIA"	12
6	Electronic AIP "CIS"	12	licenses	AIRAC (28 days)	For tablets (iOS/Windows)	6	Electronic AIP "CIS"	12
7	Paper AIP Russia (Book 2)	1	copy	subscription (amendments)	Aerodromes classes A, B, V	7	Paper AIP Russia (Book 2)	1
8	Paper AIP CIS	1	copy	subscription (amendments)	Full set	8	Paper AIP CIS	1
9	RNC chart set (paper)	1	set	as new sheets issued	Series R-17R-11, EEU+EUR+MES zone	9	RNC chart set (paper)	1
10	Onboard NDB for B-737 FMS	4	sets (per aircraft)	AIRAC (28 days)	Supply after successful testing	10	Onboard NDB for B-737 FMS	4

Highly specific aviation product references used to establish credibility

Upon analyzing the file, we observed that it contains a detailed list of aviation-related products, including navigation databases, NOTAM datasets, electronic AIP collections, and onboard systems for aircraft such as the Boeing 737. The document also specifies quantities, licensing details, update frequencies, and usage remarks, making it appear like a legitimate order summary.

The level of detail and use of industry-specific terminology suggests that this document is designed to target aviation professionals and organizations.

Technical Analysis

In this section, we will examine the technical analysis of the file extracted from the ZIP archive. The executable file which we found is named as Подтверждение заказа продукции ЦАИ.exe, which translates to "Order confirmation for CAI products.exe"

▼ PE64
 Operation system: Windows(Vista)[AMD64, 64-bit, GUI]
 Linker: Microsoft Linker(14.36.35217)
 Compiler: Rust(x86_64-pc-windows-msvc)
 Compiler: Microsoft Visual C/C++(19.36.35217)[C]
 Language: rust
 Tool: Visual Studio(2022 version 17.6)

Upon reviewing the metadata using static analysis tools, we identified that this file is a 64-bit Windows executable compiled in the Rust programming language. We then proceeded to analyze the malicious functionality of the executable, which we have divided into two stages: The first stage focuses on the activities performed by the executable on the victim's machine and the second stage covers how the executable downloads and drops a second-stage payload from a remote server depending on the threat actor, what it tries to download on the victim machines.

Stage 1 – Analysis of Malicious Executable

Upon initially looking into the malicious executable, we found that it performs executing a .docx file, which we discussed above, acting as one of a decoy, diverting victim's attention.

Decoy Document Execution

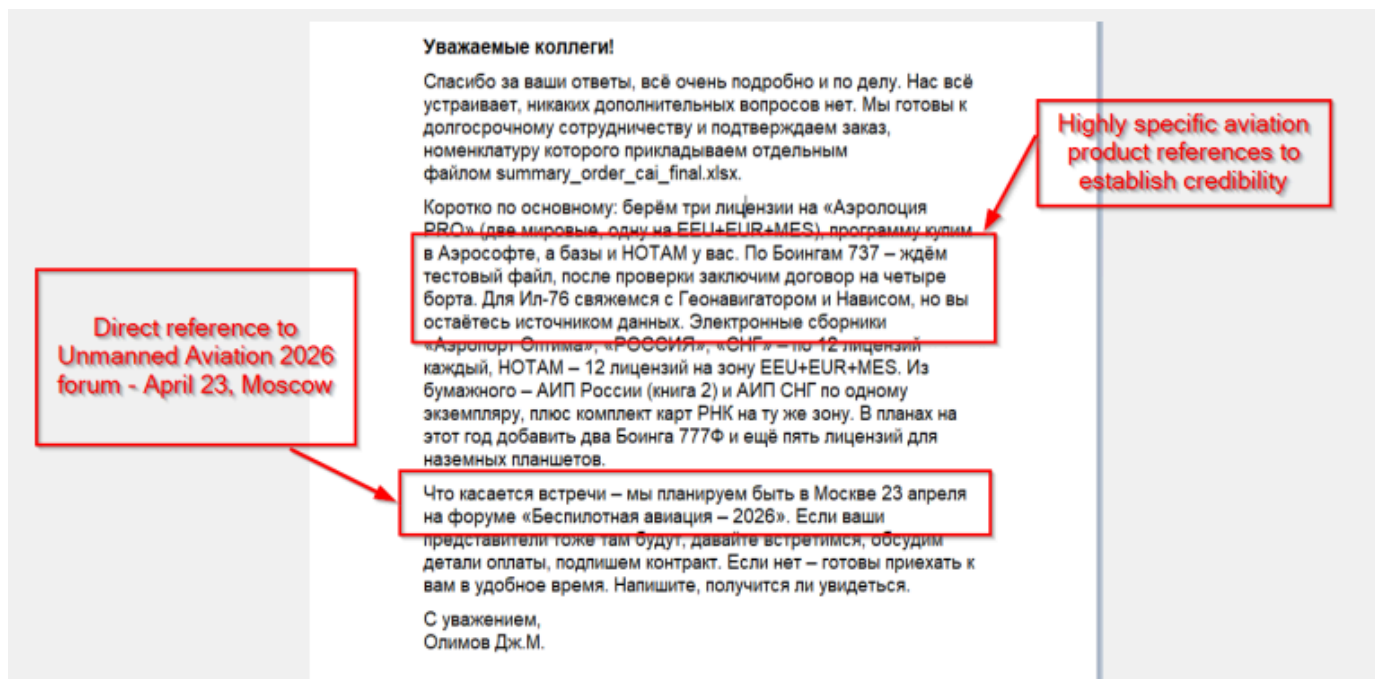
```

.rdata:000000014016AF3C db 1Fh
.rdata:000000014016AF3D db 0
.rdata:000000014016AF3E db 0
.rdata:000000014016AF3F db 0
.rdata:000000014016AF40 unk_14016AF40 db 000h ; DATA XREF: malware_main+84+0
.rdata:000000014016AF41 db 9Fh
.rdata:000000014016AF42 a0dterDenieZak db 'оdтверждение заказа продукции ЦАИ.docx' ; USERPROFILE'
.rdata:000000014016AF91 unk_14016AF91 db 0C0h ; DATA XREF: malware_main+F7+0 ; malware_main+0000+0
.rdata:000000014016AF92 db 0Ah
.rdata:000000014016AF93 db 5Ch ; \
.rdata:000000014016AF94 db 44h ; D
.rdata:000000014016AF95 db 6Fh ; o
.rdata:000000014016AF96 db 63h ; c
.rdata:000000014016AF97 db 75h ; u
.rdata:000000014016AF98 db 60h ; m
.rdata:000000014016AF99 db 65h ; e
.rdata:000000014016AF9A db 6Eh ; n
.rdata:000000014016AF9B db 74h ; t
.rdata:000000014016AF9C db 73h ; s
.rdata:000000014016AF9D db 0
.rdata:000000014016AF9E unk_14016AF9E db 43h ; C ; DATA XREF: malware_main:loc_1400020E7+0 ; malware_main+6B4A+0
.rdata:000000014016AF9F db 3Ah ;
.rdata:000000014016AFA0 db 5Ch ; \

```

Document embedded within the Executable

We observed that a document is explicitly embedded within the executable. Immediately after execution, this document is displayed on the user's screen to distract the victim and create the illusion of a benign application.



The lure document contains a business message content written in Russian, signed by “Olimov J.M.” It is designed to look like a real message from an aviation company involved in a long-term business deal.

It mentions aviation-related products such as Aerolotsia PRO licenses, NOTAM databases, AIP Russia and CIS, and aeronautical charts for aircraft like Boeing 737, IL-76, and Boeing 777F. It also includes tablet licenses. The level of detail, including license numbers and regions, makes the message appear very realistic.

It also mentions the upcoming event “Unmanned Aviation 2026” in Moscow on April 23, suggesting that the threat actor is specifically targeting Russian-speaking victims who may be attending the event.

System Fingerprinting

After executing the decoy document, the malware immediately begins collecting system information to build a unique victim profile.

```

nSize[0].m128i_u32[0] = 0;
ComputerName = GetComputerNameExW(ComputerNameNetBIOS, (LPWSTR)Buffer, (LPDWORD)nSize);
if ( (unsigned int)sub_140001F92(ComputerName) )
{
    *(_QWORD *)v678 = 0x5000000000000000uLL;
}
else
{
    if ( nSize[0].m128i_u32[0] >= 0x101uLL )
    {
        sub_14015EEE0(0, nSize[0].m128i_u32[0], 256, &off_14016B0C0);
        sub_14000B32A(v678, Buffer, nSize[0].m128i_u32[0]);
        if ( !_OFSUB_(*(_QWORD *)v678, 1) )
        {
            *(_m128i *)((char *)lpCommandLine + 8) = *(_m128i *)v678[0];
            lpCommandLine[0].m128i_i64[0] = *(_QWORD *)v678;
            LODWORD(v742[0]) = sub_14000A7C9(*(_QWORD *)v678[0], *(_QWORD *)v678[16]);
            _RAX = 0;
            __asm { cpuid }
            v83 = _RBX;
            v84 = _RCX;
            LODWORD(_RBX) = _RDX;
            nSize[0].m128i_i64[0] = 0;
            nSize[0].m128i_i64[1] = 1;
            nSize[1].m128i_i64[0] = 0;
            AdapterAddresses[0].m128i_i32[0] = _RAX;
            sub_14000D697(Buffer, AdapterAddresses, 4);
            sub_14000B32A(nSize, Buffer);
            sub_14000AF22(Buffer);
            AdapterAddresses[0].m128i_i32[0] = v83;
            sub_14000D697(Buffer, AdapterAddresses, 4);
            sub_14000B32A(nSize, Buffer);
            sub_14000AF22(Buffer);
            AdapterAddresses[0].m128i_i32[0] = v84;
            sub_14000D697(Buffer, AdapterAddresses, 4);
            sub_14000B32A(nSize, Buffer);
            sub_14000AF22(Buffer);
            AdapterAddresses[0].m128i_i32[0] = _RBX;
            sub_14000D697(Buffer, AdapterAddresses, 4);
            sub_14000B32A(nSize, Buffer);
            sub_14000AF22(Buffer);
            *(_QWORD *)v678[16] = nSize[1].m128i_i64[0];
            *(_m128i *)Buffer = nSize[0];
            v85 = sub_14000A7C9(nSize[0].m128i_i64[1], nSize[1].m128i_i64[0]);
            AdapterAddresses[0].m128i_i32[0] = 0;
            v86 = v85;
            nSize[0].m128i_i32[0] = 0;
            VolumeInformationW = GetVolumeInformationW(L"C:\\", 0, 0, (LPDWORD)nSize, (LPDWORD)AdapterAddresses, 0, 0, 0);
            v88 = (unsigned int)sub_140001F92(VolumeInformationW);
            v89 = 0;

```

The malware retrieves the machine hostname using GetComputerNameExW and the C: drive volume serial number using GetVolumeInformationW

```

v90 = sub_140014220(v89 ^ LODWORD(v742[0]) ^ v86, nSize);

```

The malware retrieves the machine hostname using GetComputerNameExW and the C: drive volume serial number using GetVolumeInformationW. These values are then XOR-combined and converted into a decimal string representation, forming a unique machine identifier for the victim's system.

Environment & Network Reconnaissance

After creating a unique ID for the victim's machine, the malware collects more information about the system and network. This step is handled by an internal module and focuses on both user and network details.

```

v742[0] = v44;
LABEL_151:
sub_1401032E0(Buffer, "USERDNSDOMAIN\\COMPUTERNAME\\windowssrc\\utilities\\collector.rs", 4);
sub_14000821B(v678, Buffer);
sub_1401032E0(Buffer, "USERDNSDOMAIN\\COMPUTERNAME\\windowssrc\\utilities\\collector.rs", 13);
sub_14000821B(lpCommandLine, Buffer);
sub_1401032E0(Buffer, "COMPUTERNAME\\windowssrc\\utilities\\collector.rs", 12);
sub_14000821B(nSize, Buffer);
sub_14000821B(Buffer, "USERDNSDOMAIN\\COMPUTERNAME\\windowssrc\\utilities\\collector.rs", 7);

```

environment variables harvested

First, the malware gathers important environment variables:

- USER – the currently logged-in username
- USERDNSDOMAIN – the domain name
- COMPUTERNAME – the system hostname
- USERPROFILE – the user's profile path

```

v34 = AdapterAddresses[0].m128i_i64[1];
AdaptersAddresses = GetAdaptersAddresses(
    2u,
    0x90u,
    0,
    (PIP_ADAPTER_ADDRESSES)AdapterAddresses[0].m128i_i64[1],
    SearchFilter);
if ( AdaptersAddresses != 111 && AdaptersAddresses != 122 )
{
    v33 = v34;
}

```

Network adapter enumeration: collects IPv4 addresses, adapter names and DNS suffixes for all available interfaces

```

v64 = 0;
--v63;
}
if ( InetNtopW(2, v62 + 2, (PWSTR)Buffer, 0x100u) )
{
    sub_14000A84A(nSize, Buffer, 256);
    v65 = nSize[0].m128i_i64[0];
    if ( v65 < 0 )
    {
        v64 = v64 + 1;
    }
}

```

Next, the malware collects network information. It scans all network adapters using GetAdaptersAddresses and retrieves their IPv4 addresses. These IP addresses are converted into readable format using InetNtopW. It also collects adapter names and DNS-related information.

All this data is then organized and prepared to be sent to the command-and-control (C2) server.

Exfiltration

The malware collects all the victim's information and converts it into a JSON format using a library called `serde_json`. It builds this JSON step by step, adding each piece of data as a key-value pair.

```

db 6Eh ; n
db 2Eh ; .
db 72h ; r
db 22h ; "
db 0
aHttpsCdnKleyma db 'https://cdn.kleymarket.ru:443Content-Typeapplication/jsonAcceptap'
; DATA XREF: sub_14000CC46+1910
db 'plication/json, text/html',0
db 0
db 0
unk_14016AF28 db 0BFh ; OFF64 SEGDEF [ rdata_14016AEBF]
; DATA XREF: malware_main+7B4310
db 0AEh
db 16h
db 40h ; @
db 1
db 0
db 0
db 0
db 0Bh
db 0
db 0
db 0
db 0
db 0

```

Hardcoded C2 endpoint and HTTP headers

After creating the JSON data, the malware encrypts it using a simple XOR method. This encrypted data is then sent to the server. After creating the JSON data, the malware encrypts it using a simple XOR method. This encrypted data is then sent to the server `cdn[.]kkeymarket[.]ru` over HTTPS (port 443) using an HTTP POST request.

Stage 2 – Second stage payload dropper

In the second stage, we will cover the malware shifts from data collection to delivering the final payload. After receiving an encrypted response from the C2 server, the malware decrypts the payload using multi-step decryption algorithms, and extracts the payload. The malware then saves the payload with a random filename inside one of the directories and executes it on the victim machine.

Payload Decryption

The encrypted payload from the C2 server is decrypted in multiple steps. The malware first extracts two values from the server response, which are used as the AES key. It then decrypts the payload using AES-256 in blocks.

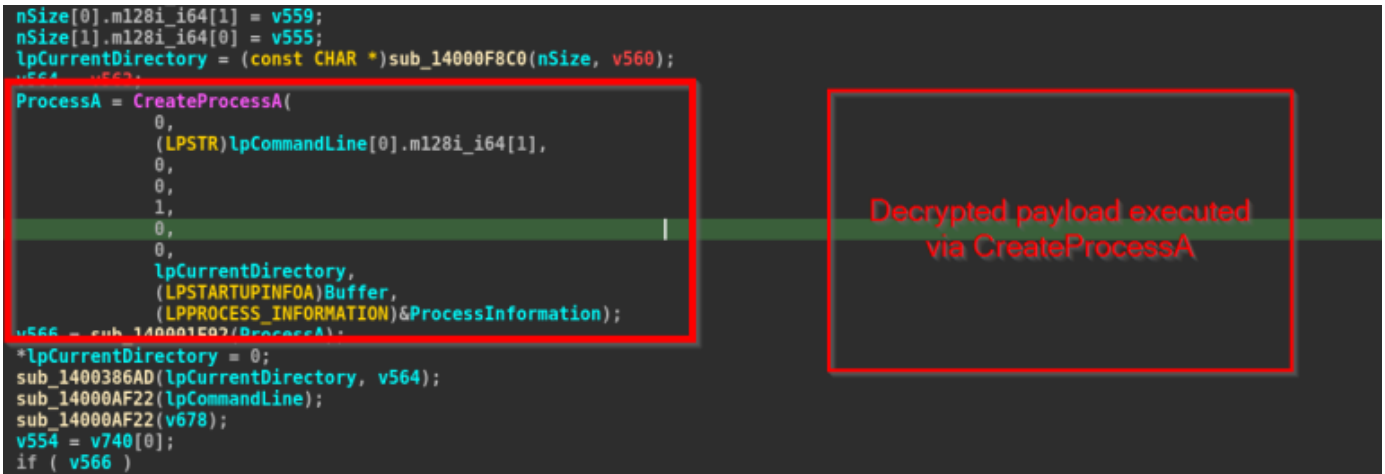
```
nSize[9] = v519;
nSize[10] = v520;
nSize[11] = v521;
nSize[12] = v522;
nSize[13] = v523;
nSize[14] = _mm_xor_si128(
    _mm_xor_si128(
        _mm_xor_si128(_mm_slli_si128(v522, 12), _mm_slli_si128(v522, 4)),
        _mm_xor_si128(_mm_slli_si128(v522, 8), _mm_shuffle_epi32(lpCommandLine[0], 255))),
    v522);
sub_14000A009(v679, nSize);
qmemcpy(v678, nSize, sizeof(v678));
v469 = 480;
v470 = (__m128i *)v678;
LABEL_B47:
qmemcpy(lpCommandLine, v470, v469);
sub_140001C6E(&v681, &v691);
qmemcpy(Buffer, lpCommandLine, sizeof(Buffer));
```

Malware creates an AES-256 encryption key using data received from the C2 server

Payload Drop & Execution

After the decryption of the payload, the malware creates a random 6-character filename using letters and numbers, and adds the .exe extension. The malware writes the payload directly to disk using `NtWriteFile`. It saves the file either in one of these locations:

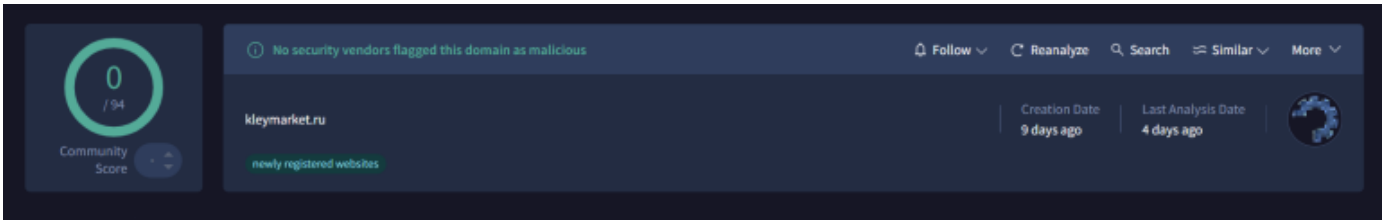
- `%USERPROFILE%\Documents\<random>.exe`
- `C:\Users\Public\Documents\<random>.exe`



Finally, the malware executes the decrypted payload using the `CreateProcessA` API. The full path to the dropped executable is passed directly as the command line, with no parent application name specified, launching a new process on the victim machine.

Now, in the next section, we will look into the infrastructure and attribution aspects of the campaign.

Infrastructure & Attribution



We analyzed the command-and-control (C2) infrastructure used in this campaign and found that the domain `hxxp://kleymarket[.]ru` was recently registered, just 9 days prior to our analysis. At the time of investigation, it was not flagged as malicious by any security vendors.

Passive DNS Replication (3)			
Date resolved	Detections	Resolver	IP
2026-04-02	1 / 94	VirusTotal	45.142.36.76
2019-11-28	0 / 94	VirusTotal	92.62.113.232
2019-09-28	0 / 94	VirusTotal	89.108.110.154

Passive DNS data reveals that the domain has resolved to multiple IP addresses over time, including:

45[.]142[.]36[.]76, 92[.]62[.]113[.]232, and 89[.]108[.]110[.]154

The most recent resolution, dated 2026-04-02, points to 45.[.]142[.]36[.]76, which appears to be the active C2 server used in this campaign. In contrast, older records from 2019 indicate that the domain was previously used for other purposes before being reused for this activity.

45.142.36.76

 AS 48347 (MTW-AS)

Summary

OSINT 0

Resolutions 5

Subdomains

DNS Rec

 Usage

Owner

MTW-AS (RU)

ASN

AS 48347 

CIDRs

45.142.36.0/24

Approximate Location

[Source](#) 

 Moscow, Moscow, RU

 55.7696° N, 37.6177° E

Further analysis using [Validin](#) confirms that the domain and its associated subdomain resolve to the IP address 45.[.]142[.]36[.]76, which appears to be the active C2 infrastructure used in this campaign. This IP is hosted under AS48347 (MTW-AS), a Russian autonomous system, with an approximate location in Moscow, Russia.

At the time of writing, we are not attributing this campaign to any known threat actor. However, the observed tactics, infrastructure, and social engineering elements suggest a well-planned and targeted operation. Given the campaign’s stealthy delivery mechanism and aviation-themed targeting, we have named this campaign as “Operation Silent Rotor”. The campaign appears to be carefully timed to align with the “Unmanned Aviation 2026” forum in Moscow, indicating a focus on aviation professionals who are likely to attend or be associated with the event.

Conclusion

SEQRITE Labs has identified a targeted spear phishing campaign affecting professionals in the Eurasian unmanned aviation sector. The campaign uses realistic aviation-related documents to gain the victim's trust, with content linked to the "Unmanned Aviation 2026" forum in Moscow. The delivered malware is a Rust-based executable that collects system information, communicates with a remote server over encrypted HTTPS, and downloads a second-stage payload for execution.

The infrastructure used in this campaign is minimal and short-lived, relying on a newly registered .ru domain. At the time of writing, we are not attributing this campaign to any known threat actor. However, the use of Russian-language lures and context-specific content suggests that the campaign is regionally focused and targets victims within the same ecosystem.

Seqrte Protection

Trojan.Win64

Indicators of Compromise (IOCs)

Hash (SHA-256)	File
5936f42ffd7fa7896eeae725b60a5d26bbf3e584712671ef5da0138ee5d58f60	Подтверждение заказа продукции ЦАИ.exe
fdef9e489f773319f55f92f712d1b7b5447d59a632b8f4173d1b161d3759ad92	cai partner (1).zip
57e26f6e3b311a1064c946b69159ee05abedf9228b2f95c65536429e7ac7fb24	cai partner.zip
a7bd8869293212e1671df90d2d41b96d4933eb9408b1111bd830e111a91bb202	Certificate of translation.PDF
2064ef387ac9e51ba72b32004d99e8a0b291dbab24ed8db30f437abf1b40cb49	Confirmation of CAICA products order and arrangement of a meeting at the Unmanned Aviation – 2026 forum to discuss payment details and sign a long-term contract.docx
89f8e42c825d09a0a50e99bbf7304d7037be33ea362a57d34f87fa7981f80126	summary_order_cai_final.xlsx

URLs

45[.]142[.]36[.]76

cdn[.]kleymarket[.]ru

MITRE ATT&CK Mapping

Tactic	Technique ID	Technique Name
--------	--------------	----------------

Initial Access	T1566.001	Phishing: Spearphishing Attachment
Execution	T1204.002	User Execution: Malicious File
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell
Execution	T1106	Native API
Defense Evasion	T1036.004	Masquerading: Match Legitimate Name or Location
Defense Evasion	T1027	Obfuscated Files or Information
Defense Evasion	T1140	Deobfuscate/Decode Files or Information
Discovery	T1082	System Information Discovery
Discovery	T1016	System Network Configuration Discovery
Discovery	T1033	System Owner/User Discovery
Discovery	T1083	File and Directory Discovery
Command and Control	T1071.001	Application Layer Protocol: Web Protocols
Command and Control	T1090.001	Proxy: Internal Proxy
Exfiltration	T1041	Exfiltration Over C2 Channel
Impact	T1105	Ingress Tool Transfer

Authors

Priya Patel

Rayapati Lakshmi Prasanna Sai

Related Posts



